ThingMagic
A DIVISION OF TRIMBLE

875-0016-07 RevA

Mercury5e and M5e-Compact Developer's Guide

For: Mercury5e and M5e-Compact (v1.7.1 and later)
USB Reader and Vega Reader

Government Limited Rights Notice: All documentation and manuals were developed at private expense and no part of it was developed using Government funds.

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose the technical data contained herein are restricted by paragraph (b)(3) of the Rights in Technical Data — Noncommercial Items clause (DFARS 252.227-7013(b)(3)), as amended from time-to-time. Any reproduction of technical data or portions thereof marked with this legend must also reproduce the markings. Any person, other than the U.S. Government, who has been provided access to such data must promptly notify ThingMagic, Inc.

ThingMagic, Mercury, Reads Any Tag, and the ThingMagic logo are trademarks or registered trademarks of ThingMagic, Inc.

Other product names mentioned herein may be trademarks or registered trademarks of ThingMagic, Inc. or other companies.

ThingMagic
A Division of Trimble
One Cambridge Center, 11th floor
Cambridge, MA 02142.
866-833-4069

07 Revision A
January, 2012

Revision Table

Date	Version	Description
2/2008	Rev1	Separated out M4e content and Reader Assistant into standalone documents. Created new M5e Family DevGuide
2/2008	Rev1	Updated TX Power section to correct TX Write power note.
2/2008	Rev1	Fixed Send CW Signal content with correct value for PRBS signal.
3/2008	Rev1	Added input voltage details to M5e/C Hardware comparison table.
3/2008	Rev1	Added FCC and IC Regulatory Statements
3/2008	Rev1	Updated timestamp field in Tag Read Meta Data description to indicate its ticks from read command invocation.
3/2008	Rev1	Added details on new functionality for release 1.0.37: • M5e EU info (Hardware Support) • PRC (China) region support (get/set Current Region) • Support for user control of Gen2 Q value (Get/Set Protocol Parameter) • LBT on/off and LBT Threshold setting in supported regions (Get/Set Current Region) and LBT default settings in Regulatory Support section. • Custom NXP Silicon Commands (Tag Specific) • Get Reader Statistics • Get tag read metadata with Read Tag Single • Get tag read metadata with Read Tag Data • Read Tag Multiple with Tag Singulation • Read Tag Multiple with embedded commands • Updates to Tag Select functionality
4/2008	Rev1	Added Gen2 Memory Map and algorithm information to Tag Singulation section.
4/2008	Rev2	Added information on Transmit mode power consumption to Transmit Modes
4/2008	Rev2	Corrected NXP command: • names: changed ‘Quiet’ to ‘ReadProtect’ • EAS Alarm and Calibrate syntax

Date	Version	Description
6/2008	Rev3	• Fixed Arbser help info to include -l5a for loading M5e firmware • Fixed Get Tag Buffer Response Fields to show Read Count as conditional
9/2008	02 RevA	• Added info on EU2 usage and caution • Update Get/Set Reader Configuration commands to reflect the new key/value pair format • Fixed the CRC calc sample code • Fixed PRC region frequency range
11/2008	02 RevA	• added info about why use Write Tag EPC over Write Tag Data
4/2009	03 RevA	• Added 1.1.1 features
6/2009	04 Rev1	• fixed response fields - missing metadata field - in Get Tag Buffer
8/2009	04 Rev1	• Fixed Read Tag Data meta data fields • KR2 region support info • misc doc bug fixes
12/2010	05 Rev1	• Added detail to Tag Singulation/Select section • Updated RS232 electical specs • Added MercuryAPI info • updated copyright and FCC info with new company info • added note to Lock Tag
3/2011	06 Rev1	Updates for firmware v1.5.1 • New Custom commands support • more data returned for embedded tag read data operations.
1/2012	07 RevA	Updates for firmware 1.5.2 • Set Gen2 Write Response Time (under 0x9B) • IDS SL900A Gen2 Tag Specific (2Dh) support

Communication Regulation Information

EMC FCC 47 CFR, Part 15
Industrie Canada RSS-210

Federal Communication Commission Interference Statement

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

FCC Caution: Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment.

This transmitter module is authorized to be used in other devices only by OEM integrators under the following conditions:

1. The antenna(s) must be installed such that a minimum separation distance of 25cm is maintained between the radiator (antenna) & user’s/nearby people’s body at all times.

2. The transmitter module must not be co-located with any other antenna or transmitter.

As long as the two conditions above are met, further transmitter testing will not be required. However, the OEM integrator is still responsible for testing their end-product for

any additional compliance requirements required with this module installed (for example, digital device emissions, PC peripheral requirements, etc.).

Note

In the event that these conditions can not be met (for certain configurations or co-location with another transmitter), then the FCC authorization is no longer considered valid and the FCC ID can not be used on the final product. In these circumstances, the OEM integrator will be responsible for re-evaluating the end product (including the transmitter) and obtaining a separate FCC authorization.

The OEM integrator has to be aware not to provide information to the end user regarding how to install or remove this RF module in the user manual of the end product.

User Manual Requirement

The user manual for the end product must include the following information in a prominent location;

"To comply with FCC's RF radiation exposure requirements, the antenna(s) used for this transmitter must be installed such that a minimum separation distance of 25cm is maintained between the radiator (antenna) & user's/nearby people's body at all times and must not be co-located or operating in conjunction with any other antenna or transmitter."

AND

"The transmitting portion of this device carries with it the following two warnings:

"This device complies with Part 15...."

AND

"Any changes or modifications to the transmitting module not expressly approved by ThingMagic Inc. could void the user's authority to operate this equipment" "

End Product Labeling

The final end product must be labeled in a visible area with the following:

"Contains Transmitter Module FCC ID: QV5MERCURY5E"

or

“*Contains FCC ID: QV5MERCURY5E.”*

(Replace QV5MERCURY5E, for the Mercury5e, with QV5MERCURY5EC for the M5e-Compact).

Industry Canada

Operation is subject to the following two conditions: (1) this device may not cause interference, and (2) this device must accept any interference, including interference that may cause undesired operation of the device.

To reduce potential radio interference to other users, the antenna type and its gain should be so chosen that the equivalent isotropically radiated power (e.i.r.p.) is not more than that permitted for successful communication.

This device has been designed to operate with the antennas listed in Authorized Antennas. Antennas not included in this list are strictly prohibited for use with this device.

To comply with IC RF exposure limits for general population/uncontrolled exposure, the antenna(s) used for this transmitter must be installed to provide a separation distance of at least 25 cm from all persons and must not be collocated or operating in conjunction with any other antenna or transmitter.

End Product Labeling

The final end product must be labeled in a visible area with the following:

“Contains ThingMagic Inc. M5e-Compact (or appropriate model number you’re filing with IC) *transmitting module FCC ID: QV5MERCURY5EC (IC: 5407A-MERCURY5EC)”*

Contents

Introduction

The ThingMagic® Mercury® embedded modules are RFID engines that you can integrate with other systems to create RFID-enabled products.

Applications to control the M5e-Family of modules and derivative products can be written in the low level Serial Protocol Command Set and also using the high level MercuryAPI.

The MercuryAPI supports Java and .NET environments starting with version 1.1 and C starting with version 1.5. The MercuryAPI Software Development Kit (SDK) contains sample applications and source code to help developers get started demoing and developing functionality. For more information on the MercuryAPI see the *MercuryAPI Programmers Guide* and the *MercuryAPI SDK*, available on the ThingMagic website.

For assistance using thelow level Serial Protocol Command Set a comprehensive user interface called the *Reader Assistant* provides screens to configure the reader, and read from and write to tags and displays the resulting serial commands in its *Serial Log*. In addition, there are screens for updating firmware and debugging. For those communications that cannot be provided by the *Reader Assistant*, source code for the *ArbSer* application (available upon request from support@thingmagic.com) is available. *ArbSer* is a terminal program with which you can communicate with the modules. *ArbSer* uses commands that are detailed in this document. See Command Set.

This document is for developers and explains how to incorporate the Mercury5e (M5e) or M5e-Compact product within a third-party host system.

Note

For an overview of the Developer's Kit hardware and initial setup information see Appendix B: Getting Started - Devkit

Product Line Overview

The embedded modules were designed to be incorporated into products requiring powerful RFID capabilities in a small form factor.

The M5e is a small form-factor, low power, low cost Gen2 module. The M5e is ideal for embedding a powerful RFID module with read and write capabilities into a product or system.

The M5e-Compact is a smaller version of the M5e. It has one MMCX connector for a monostatic antenna. The M5e-Compact is ideal for use in hand-held printers and other applications where size is the highest priority.

Product Specifications

The following table compares and contrasts the two Mercury embedded modules:

Comparison of features between M5e, and M5e-Compact

Item	M5e	M5e-Compact
Processor	Atmel AT91SAM7S-256	Atmel AT91SAM7S-256
Flash memory	256 kB	256 kB
On-chip RAM	64 kB	64 kB
RF Architecture	ASIC Impinj Indy1000 with ThingMagic front end for improved sensitivity	ASIC Impinj Indy1000 with ThingMagic front end for improved sensitivity
Input Power Require-ments	+5VDC +/-4% 7.5W max (6.5 typical) 25mV max peak-peak ripple no spectral spike greater than 5mVpp in any 1kHz band	+3 to +5.5VDC 2.7W max (2.6 typical) 25mV max peak-peak ripple no spectral spike greater than 5mVpp in any 1kHz band
Protocols supported	GEN2, ISO 18000-6C	GEN2, ISO 18000-6C
Dimensions:	82mm L x 54mm W x 5mm H	56mm L x 35.6mm W x 5mm H
Regions supported	NA, EU (3 variations), Korea, India, China	NA, EU (3 variations), Korea, India, China

Electrostatic Discharge	• IEC-61000-4-2 discharge direct to operational antenna port tolerates max 300 Volt Pulse • MIL-883 3015.7 discharge direct to operational antenna port tolerates max 1200 Volt Pulse Note: Survival level varies with antenna return loss and antenna characteristics. See Appendix E: Environmental Considerations for methods to increase ESD tolerances.

W A R N I N G !

The M5e antenna ports may be susceptible to damage from Electrostatic Discharge (ESD). Equipment failure can result if the antenna or communication ports are subjected to ESD. Standard ESD precautions should be taken during installation and operation to avoid static discharge when handling or making connections to the M5e reader antenna or communication ports. Environmental analysis should also be performed to ensure static is not building up on and around the antennas, possibly causing discharges during operation.

Hardware Overview

The following sections explain the M5e and M5e-Compact hardware, and their hardware revisions.

Note

The M5e and M5e-Compact are the same except where specified that they are different.

M5e and M5e-Compact Hardware

The M5e and the M5e-Compact are single board modules designed for more space-constrained applications. The digital and analog electronics are on the same circuit board. They use a custom RFID reader chip or Application Specific Integrated Circuit (ASIC).

Microcontroller

The M5e and M5e-Compact have Atmel ARM7 microcontrollers with 256 kB of on-chip flash memory for storage of all calibration and program data.

RFID ASIC

All base-band analog circuitry and PLL circuitry are contained within the Impinj Indy1000 RFID ASIC with ThingMagic front end for improved sensitivity.

Connectors

The M5e supports two MMCX connectors for bistatic or monostatic antennas. The M5e-Compact supports one MMCX connector for a monostatic antenna.

Hardware Revisions

Table 2 lists the different Mercury Embedded hardware versions and their power output capabilities.

Mercury Embedded Hardware Versions

Module	Minimum Power Out	Maximum Power Out	Comments
M5e 1 Watt	5 dBm	30 dBm	Supports 1 Watt operation in NA and KR regions and less than 0.2 Watt operation in EU and PRC regions in typical applications.
M5e EU 1 Watt	5 dBm	30 dBm	Supports 1 Watt operation in EU region.
M5e-Compact 0.2 Watt	10 dBm	23 dBm	Supports 0.2 Watt operation in all regions.

M5e/M5e-Compact Digital Connectors

The digital connector provides power, serial communications signals, and access to the GPIO inputs and outputs.

M5e and M5e-Compact Connector

The communications interface for the M5e and M5e-Compact is a 12-pin digital connector. This connector provides power, serial communications signals, and access to the GPIO inputs and outputs. See the following table:

Pin-out of 12-pin Digital Communications Connector

Pin #	Signal
1	+5V
2	+5V
3	GND
4	GND
5	Digital Output 1
6	Digital Output 2
7	Digital Input 1
8	Digital Input 2
9	UART TTL RX from host
10	UART TTL TX to host
11	Mfg test purposes
12	Mfg test purposes

SW Overview

The software (SW) for the embedded products consists of two separate programs that coexist in flash memory:

- The boot loader, which is started at power on, is not field upgradable. It is programmed into flash when the module is manufactured.
- The application firmware, which implements the actual reader functionality, is field upgradable.

Boot Loader

The boot loader provides low-level functionality. This program provides a customer interface for upgrading the application firmware and storing data into flash.

When a module is powered up or reset, the boot loader code is automatically copied from sector 0 of flash into the Microprocessor's on-chip RAM, and executed. The boot loader provides the following features:

- Ability to read / write / erase flash memory
- Upgrade application FW
- Change serial baud rate
- Verify image CRC

Application Firmware

The application firmware (FW) is an important software component of the module. It contains the protocol code as well as all the user interfaces to set and get various system parameters. The application FW is started using the **Boot Firmware** command in the boot loader; it does not start by itself upon power up.

Note

You can use the Reader Assistant to upgrade the reader firmware through the bootloader.

Verifying Application FW Image CRC

The application FW has an image level Cyclic Redundancy Check (CRC) embedded in it to protect against corrupted firmware during an upgrade process. (If the upgrade is unsuccessful, the CRC will not match the contents in flash.) When the boot loader starts the application FW, it first verifies that the image CRC is correct. If this check fails, then the boot loader does not start the application FW.

The upgrade process uses a series of individual 250-byte packet write operations to ensure that an upgrade is successfully completed for the complete image. It also ensures that the application FW in flash was not corrupted accidently, and can be expected to perform properly when executed.

About the Reader Assistant

An easy-to-use user interface (*Reader Assistant*) can be installed to simplify reader communication. This *Reader Assistant* can be used to demonstrate the embedded module or perform detailed evaluations of the product's performance. The *Reader Assistant* has the following features:

- Real-time logging of all serial transmits and receives with a timestamp
- Reading and writing of all tag commands
- Debugging capability
- Reading, writing, and modifying data stored in flash memory
- Reading and writing to applications stored in flash memory
- Updating of new firmware releases
- Setting and getting parameters

Application Development

See Appendix B: Getting Started

Functionality of the Embedded Modules

This section highlights some of the functionality of the modules. The details for using the serial commands to control this functionality are found in Overview of the Communication Protocol.

Regional Support

The modules have differing levels of support for operation and use under the laws and guidelines of several regions. The regional support is shown in the following table.

Supported Regions

Region	Regulatory Support	M5e	M5e-C	M5e EU
North America (NA)	FCC 47 CFG Ch. 1 Part 15 Industrie Canada RSS-210	Yes	Yes	No
European Union (EU)	ETSI EN 302 208	Yes	Yes	Yes
European Union (EU2)	ETSI EN 300 220	Yes	Yes	Yes
European Union (EU3)	Revised ETSI EN 302 208	Yes	Yes	Yes
Korea (KR)	MIC	Yes	Yes	No
Korea (KR2)	KCC (2009)	Yes	Yes	No
India (IN)	Telecom Regulatory Authority of India (TRAI), 2005 regulations	Yes	Yes	Yes
[1]People's Republic of China (PRC)	SRRC, MII	Yes	Yes	No
Australia (AU)	ACMA LIPD Class Licence Variation 2011 (No. 1)	Yes	Yes	No
New Zealand (NZ)	Radiocommunications Regulations (General User Radio Licence for Short Range Devices) Notice 2011	Yes	Yes	No
Open Region	No regulatory compliance enforced	Yes	Yes	Yes

Note: 1- Under the current PRC regulations the module alone should berestricted to an output power setting of +21dBm or below. However, the exact maximum power setting is determined when the device in which the module is installed, and the module, is tested as a system.

The regional functionality is set using a single serial command, Set Current Region (97h). Setting the Region configures the regional default settings including:

- Loads the Frequency Hop Table with the appropriate table for the selected region.
- Sets the PLL frequency to the first entry in the hop table, even if the RF is off.
- Selects the transmit filter, if applicable.
- Executes the Listen Before Talk (LBT) algorithm in supported regions, using the defaults specified below or the user specified values when calling Set Current Region (97h).

Regions Supporting LBT - Default Settings

Region	LBT Enabled	LBT Threshold
EU	Yes	-96 dBm
EU3	No	n/a
Open	No	-96 dBm

Note

The **Open Region** allows the module to be manually configured within the full capabilities supported by the hardware. No regulatory limits, including: frequency range, channel spacing and transmit power limits, are enforced. The Open Region should be used with caution.

C A U T I O N !

The EU2 Region **does not implement LBT. Therefore, the limits in section 8.10 of the EN 300 220 specification, referring to duty cycle limits, apply. Since the module does not enforce any duty cycle or power limitations when configured to use the EU2 region and will transmit 100% of the time during a tag operation it is the responsibility of the user to appropriately manage the RF duty cycle and power levels to conform with EN 300 220.**

Frequency Setting

The modules have a PLL synthesizer that sets the modulation frequency to the desired value. Whenever the frequency is changed, the module must first power off the modulation, change the frequency, and then turn on the modulation again. Since this can take several milliseconds, it is possible that tags are powered off during a frequency hop. In addition to setting the default regional settings, the modules have commands that allow the transmit frequency to be set manually.

C A U T I O N !

Use these commands with extreme caution. It is possible to change the module's compliance with the regional regulations.

Frequency Units

All frequencies in the Mercury embedded products are expressed in kHz using unsigned 32-bit integers. For instance, a carrier frequency of 915 MHz is expressed as 915000 kHz.

The PLL is set automatically to the closest frequency - based on the minimum frequency quantization for the current region - that matches the specified value. The Mercury embedded modules have an absolute minimum quantization of 25 kHz. Each region also has a minimum quantization based on regulatory specifications, which may be greater. The following table details the frequency quantization in kHz for each region setting.

Regional Frequency Quantization

Region	Frequency Quantization	Minimum Frequency	Maximum Frequency
NA	250 kHz	902,000 kHz	928,000 kHz
EU	100 kHz	865,100 kHz	867,900 kHz
IN	100 kHz	865,200 kHz	866,800 kHz
EU2	50 kHz	869,000 kHz	869,850 kHz
EU3	100 kHz	865,600 kHz	867,600 kHz
KR	25 kHz	910,000 kHz	914,000 kHz
KR2	25kHz	917,300 kHz	920,300 kHz
PRC	250 kHz	920,125 kHz	924,875 kHz
AU	250kHz	920,750 kHz	925,250 kHz
NZ	250 kHz	922,250 kHz	927,250 kHz
Open	25 kHz	860,000 kHz	960,000 kHz

When manually setting frequencies the module will round down for any value that is not an even multiple of the supported frequency quantization.

For example: In the NA region, setting a frequency of 902,999 kHz results in a setting of 902,750 kHz.

When setting the frequency of the module, any frequencies outside of the valid range for the specified region are rejected.

Frequency Hop Table

The frequency hop table determines the frequencies used by the modules when transmitting. The hop table characteristics are:

- Contains up to 62 slots.
- Valid frequencies for the region currently selected.
- Changes not stored in flash, thus changes made are not retained after a power cycle or a restart of the boot loader.
- Inability to change individual entries after uploading without reloading the entire table.

- Frequencies used in the order of entries in the table.

If necessary for a region, the hop table can be randomized to create a pseudo-random sequence of frequencies to use. This is done automatically using the default hop tables provided for each region.

EU3 Region

By default the frequency hop table for the EU3 region will use four channels. In addition to the default configuration the EU3 region can also be used in a single channel mode. These two modes of operation are defined as:

Single Channel Mode

Set by manually setting the frequency hop table to a single frequency. In this mode the module will occupy the set channel for up to four seconds, after which it will be quiet for 100ms before transmitting on the same channel again.

Multi Channel Mode

Set by leaving the default or manually setting more than one frequency in the hop table. In this mode the module will occupy one of the configured channels for up to four seconds, after which it may switch to another channel and immediately occupy that channel for up to four seconds. This mode allows for continuous operation.

PRC Region

The PRC specifications limits channels 920 to 920.5MHz and 924.5 to 925.0MHz to transmitting at 100mW or below. The default hoptable on the Mercury5e and M5e-Compact uses only the center channels which allow 2W ERP, 1W conducted, power output. If the hoptable is modified to use the outer, lower power channels the RF level will be limited to the outer channels limit, 100mW or +20dBm

KR2 Region

The first frequency channel (917,300kHz) of the KR2 region will be derated to +27dBm to meet the new Korea regulatory requirements. All other channels operate up to +30dBm. In the worst case scenario, each time the derated channel is used it will stay on that channel for 400ms. The fastest it will move to the next channel, in the case where no tags are found using that frequency, it will move to the next channel after 10 empty query rounds, approximately 120ms.

RF Power Setting

The power setting is calibrated at the factory and parameters are stored in flash memory to ensure the power output is within +/– 1 dB of the desired setting. The power limits are set by both hardware limitations and enforced by the firmware.

Power Units

All power values are reported in centi-dBm. Therefore, a power setting of 2500 corresponds to 25 dBm. All power values in the serial interface are specified as unsigned 16-bit integers. A dBm means power referenced to 1mW. Therefore, the conversion from watts to dBm is:

$$dBm = 10 \log_{10}(\text{power in mW})$$

For example: 0.1W = 100mW = 20dBm and 1W = 1000mW = 30dBm.

Power Calibration

A power calibration event occurs when the Power set point is changed, the Frequency is changed, or the RF field is turned on. The power calibration routine calibrates the power within 10 ms.

Note

Power calibration only occurs once when the RF power is turned on. It does not occur periodically when the RF field is on.

This is not an issue during normal operation, since a frequency hop occurs at least every 400 ms and thermal drift does not affect the power level significantly. However, in CW waveform mode (**Transmit CW Signal**), no power calibration occurs unless the power or frequency is changed. Thus, it is possible to experience thermal drift in this usage if the unit is left to transmit continuously for a significant period of time.

TX Read Power

The TX read power is used for all non-write commands to tags. This may include, but is not limited to, commands to read a tag ID and read tag data.The TX Read Power can be set in two ways:

- The default, module-wide, setting is specified using Set Read TX Power (92h). This setting is used for all read operations on all ports, except:
- The power for individual logical antennas can be specified using Set Antenna's Power and Settling Time. When a logical antennas power is set using that command it will override the module-wide setting for all read operations.

TX Write Power

The TX write power is used only during commands that change a tag's state. This includes commands for **Lock**, **Kill**, as well as **Tag ID Write** and **Tag Data Write**.

Note

When performing write operations as part of Tag Inventory With Embedded Operations the TX Read Power will be used for those write operations, NOT the TX Write Power.

The TX Write Power can be set in two ways:

- The default, module-wide, setting is specified using Set Write TX Power (94h). This setting is used for all write operations on all ports, except:
- The power for individual logical antennas can be specified using Set Antenna's Power and Settling Time. When a logical antennas power is set using that command it will override the module-wide setting for all write operations.

Note

During Write commands, which both write to and verify the written contents, the entire operation will be done at the Tx Write Power, including tag singulation and the write verify.

Antenna Ports

The modules have two antenna ports, except for the M5e-Compact which has one. While each port is capable of both transmitting and receiving, only one port can transmit or receive at a time. The antenna ports can be configured for either monostatic or bistatic mode using the **Set Antenna Port** command Set Antenna Port (91h) or selecting the antenna port from the *Reader Assistant* Antenna Port menu Configuring the Reader.

In addition, the modules support Using a Multiplexer, allowing up to eight logical antenna ports, controlled using the two GPOutput lines and the internal physical port J1/J2 switching.

Note

The M5e-Compact has one antenna port and only supports monostatic operation.

Monostatic Mode

To set up the module to use a single antenna in monostatic mode, connect the antenna to port 1 (labeled J1 on Printed Circuit Board) which is responsible for both TX and RX communication. If possible, terminate the unused port to prevent damage to the reader if it is accidentally commanded to transmit through it. The communication is configured using the **Set Antenna Port** command by setting both the TX and RX antenna ports to the same value.

To use two antennas in monostatic mode, connect one monostatic antenna to port 1 and the other antenna to port 2. With two monostatic antennas connected you can choose to manually select which antenna to use (based on the antenna set with Set Antenna Port) for tag operations or allow the module to search on both antennas automatically. Multi-antenna Search can be specified by the Antenna Flag in the Read Tag Multiple (22h) command.

When using LBT, it will always be performed on antenna port 2 in monostatic mode. Due to this requirement you should always insure to have an antenna connected to port 2 when using a region with LBT enabled.

Bistatic Mode

In bistatic mode, antenna port 1 transmits data and antenna port 2 receives data. This requires either a two-port antenna, or two single-port antennas. This configuration provides better isolation between the transmitter and receiver, and should be used whenever possible.

Note

For bistatic mode, port 1 must be the TX antenna, and port 2 must be the RX antenna.

Using a Multiplexer

Multiplexer switching is controlled through the use of the internal module physical port J1/J2 switch along with the use of one or both of the General Purpose Inputs/Outputs (GPIO)

Output lines. In order to enable automatic multiplexer port switching the module must be configured to use *Use GPIO as Antenna Switch* in Set Reader Configuration(9Ah).

Once the GPIO line usage has been enabled the following control line states are applied when the different Logical Antenna settings are used in Set Antenna Port (91h).

Note

The TX/RX Logical Antenna values are static labels indicating the available control line states. The specific physical antenna port they map to depends on the control line to antenna port map of the multiplexer in use. The translation from TX/RX Logical Antenna label to physical port must be maintained by the control software.

GPIO 1 & 2 Used for Antenna Switching

TX/RX Logical Antenna Setting	GPIO Output 1 State	GPIO Output 2 State	Physical Module Port State
01 01	Low	Low	TX=1 RX=1
02 02	Low	Low	TX=2 RX=2
03 03	Low	High	TX=1 RX=1
04 04	Low	High	TX=2 RX=2
05 05	High	Low	TX=1 RX=1
06 06	High	Low	TX=2 RX=2
07 07	High	High	TX=1 RX=1
08 08	High	High	TX=2 RX=2
01 02	Low	Low	TX=1 RX=2
03 04	Low	High	TX=1 RX=2
05 06	High	Low	TX=1 RX=2
07 08	High	High	TX=1 RX=2

Note

Just as under non-Multiplexing operation, Physical Module Port #1 must be used for transmitting and Port #2 for receiving when configured for bistatic operation.

If only one GPIO Output line is used for antenna control, the combinations of the available output control line states (the GPIO line in use and the module port) result in a subset of TX/RX logical antenna settings which can be used:

ONLY GPIO 1 Used for Antenna Switching

TX/RX Logical Antenna Setting	GPIO Output 1 State	Physical Module Port State
01 01	Low	TX=1 RX=1
02 02	Low	TX=2 RX=2
05 05	High	TX=1 RX=1
06 06	High	TX=2 RX=2
01 02	Low	TX=1 RX=2
05 06	High	TX=1 RX=2

ONLY GPIO 2 Used for Antenna Switching

TX/RX Logical Antenna Setting	GPIO Output 2 State	Physical Module Port State
01 01	Low	TX=1 RX=1
02 02	Low	TX=2 RX=2
03 03	High	TX=1 RX=1
04 04	High	TX=2 RX=2
01 02	Low	TX=1 RX=2
03 04	High	TX=1 RX=2

Power Management

The modules use different methods and levels of power management.

Power Modes

The M5e and M5e-Compact were designed for power efficiency and offer several different power management modes. The following lists the current modes being offered:

- Full Power Mode – In this mode, the unit operates at full power to attain the best performance possible. This mode is only intended for use in cases where power consumption is not an issue. This is the default Power Mode at startup.
- Minimal Saving Mode – This automatically executes basic power savings that do not severely degrade system performance May result in a nominal 1 ms additional delay on the M5e and 10 ms of delay on the M5e-Compact.
- Medium Saving Mode – This mode may add up to 50 ms of delay on both the M5e and the M5e-Compact between commands. It performs more aggressive power savings, such as automatically shutting down the analog section between serial commands, and then restarting it whenever a tag command is issued.
- Maximum Saving Mode – This mode essentially shuts down the digital and analog boards, except to power the bare minimum logic required to wake the processor. It can take up to 150 ms on the M5e and up to 100 ms on the M5e-Compact to wake the processor and execute the desired command.

Note

Maximum Saving Mode only supports communications at 9600 baud

Transmit Modes

The Transmit Modes (set through the Set Reader Configuration(9Ah) command) options apply only to the M5e.

High Performance Mode

This mode maximizes performance without regard to thermal characteristics or power consumption. High performance mode consumes up to 7.5W (6.5 typical) of DC power at any RF setting. The is the default setting on the M5e.

Low Power Mode

This mode minimizes power consumption and optimizes thermal characteristics without compromising performance in many, but not all, applications. Low power mode uses less DC power than High Performance mode at all RF transmit levels. It consumes up to 7.5W (6.5 typical) at a transmit level of +30dBm, decreasing to 3.5W at transmit levels of +20dBm and below. Low Power mode should be selected for most applications, including:

- All tag printer applications (where tags are placed very close to the antenna)
- All applications where the Read and Write transmit power levels are between +27 and +30 dBm.

If the application does not meet the above criteria, the optimum mode can be determined by considering a combination of conditions: expected transmit on/off duty cycle, expected ambient temperature, and whether the reader will be operating in a noisy RF environment.

The following table provides recommendations on which Power Mode to use for various application environments: printer, few readers (non-dense reader environment), many readers (dense reader environment); based on the required RF Power level.

Power Mode per Operational Environment and RF Power Setting

RF Transmit Power	5-18 dBm	18-27dBm	27-30 dBm
Low Power Mode	• Printer • Few Readers	• Printer	• Printer • Few Readers • Many Readers
High Performance Mode	• Many Readers	• Few Readers • Many Readers	•

Thermal characteristics are another differentiator between the two modes. Low Power Mode allows the module to operate in a 60°C ambient environment, regardless of how continuously the module is transmitting. High Performance mode is only recommended in a 60°C environment if the module is actively transmitting less than 50% of the time. If the module is transmitting more often than this, a lower maximum ambient temperature must be maintained as show in the following table:

Power Mode per Duty Cycle and Ambient Temperature

Transmit Duty Cycle (% transmitting)	50%	75%	100%
Low Power Mode	60°C	60°C	60°C
High Performance Mode	60°C	55°C	40°C

Tag Buffer

The Tag buffer stores tags, and their metadata, found using the Read Tag Multiple (22h) command. The size of the tag buffer for each module is defined in the following table:

Tag Buffer Size

	M5e	M5e-Compact
Tag Buffer Size in Tag ID entries	200	200

Each tag entry consists of a fixed number of bytes. The size depends on the value set for the Max EPC Length parameter in Set Reader Configuration(9Ah). Each entry consists of the following fields:

Tag Buffer Entry

Total Entry Size	Field	Size	Description
18 bytes (Max EPC Length = 96bits)	EPC Length	2 bytes	Indicates the actual EPC length of the tag read. Cannot exceed the Max EPC length setting.
	PC Word	2 bytes	Contains the Protocol Control bits for the tag.
	EPC	12 bytes	Contains the tag's EPC value padded with trailing zeros if the size is less than the Max EPC Length size.
	Tag CRC	2 bytes	The tag's CRC.
68 bytes (Max EPC Length = 496bits)	EPC Length	2 bytes	Indicates the actual EPC length of the tag read. Cannot exceed the Max EPC length setting.
	PC Word	2 bytes	Contains the Protocol Control bits for the tag.
	EPC	62 bytes	Contains the tag's EPC value padded with trailing zeros if the size is less than the Max EPC Length size.
	Tag CRC	2 bytes	The tag's CRC.

In addition to the tag EPC data each entry contains meta data about how, where and when the tag was read. When using the Get Tag Buffer (29h) command you can choose to get the following tag meta data returned with each tag extracted from the tag buffer:

Tag Read Meta Data

Meta Data Field	Description
Antenna ID	The antenna on with the tag was read. If the same tag is read on more than one antenna there will be a tag buffer entry for each antenna on which the tag was read. When Using a Multiplexer, if appropriately configured, the Antenna ID entry will contain the logical antenna port of the tag read.
Read Count	The number of times the tag was read on [Antenna ID].
Protocol ID	Protocol of tag. Only 0x05 (Gen2) is supported.
Timestamp	The time the tag was read, relative to the time the command to read was issued, in milliseconds. If the Tag Read Meta Data is not retrieved from the Tag Buffer between read commands there will be no way to distinguish order of tags read with different read command invocations.
Tag Data	When Read Tag Multiple (22h) with an embedded Read Tag Data (28h) is called the Tag Buffer will contain up to 32 bytes of data returned for each tag entry. Note: Tags with the same TagID but different Tag Data can be considered unique and each get a Tag Buffer entry if set in Set Reader Configuration(9Ah). By default it is not.
Frequency	The frequency on which the tag was read
RFU	Reserved for Future Use - ThingMagic Only
LQI/RSSI	The receive signal strength of the tag response.

Whenever a Tag entry is placed in the buffer, it uses up a single entry with the EPC section containing the *maximum EPC length* number of bits, regardless of the actual EPC size of the tag read. The extra bits in the entry are padded with trailing zeros.

After the **Read Tag Multiple** command finishes, it places all of the found tags into the Tag buffer, and then returns the number of tags found to the user. Only unique tags read on each antenna are added to the Tag buffer; none of the entries show repeated Tag EPCs, except when the same tag is read by both antennas, although repeated reads on an antenna will cause the Read Count field to be incremented for that tag entry. Multiple **Get Tag Buffer** commands must be sent to read out the Tags. The Tag buffer acts as a First

In First Out (FIFO) — the first Tag found by the reader is the first one to be read out. See Get Tag Buffer (29h).

The Tag buffer is reset when the Clear Tag Buffer (2Ah) command is sent or when the protocol is set or changed using Set Current Tag Protocol (93h). This allows multiple **Read Tag Multiple** commands to be used to acquire one consistent tag buffer set.

Tag Operations

In addition to inventorying tags the M5e supports commands, defined in Application Tag Commands, which operate on individual tags in a variety of ways. The various operations can be performed on the first tag to respond, the default behavior for all but Read Tag Multiple (22h), as well as on all tags in the field as part of a Tag Inventory With Embedded Operations. Most of the operations can be performed with Tag Singulation/Select Functionality allowing them to be applied to a specific tag or set of tags.

Custom Tag Commands

Many tags now support non-"Gen2 Standard" operations. Features such as read locking memory, setting digital outputs, reading sensor data - such as temperature - from sensor tags among many others. The supported commands are listed in the Gen2 Tag Specific (2Dh) commands as well as in the *MercuryAPI Programmers Guide | Level 2 | Advanced Tag* Operations section.

Flash Memory

The modules have on-board flash memory. This flash is divided into four different sectors of varying sizes. Table 4 shows the memory map for the modules. Only sector 0x03 is set aside for user data and the other sectors are used by the application FW. For cross-platform and legacy development purposes two methods of accessing flash on the M5e-family modules is supported, one based on the memory map of the M4e. The memory map for the M4e is very different than the memory map for the M5e and M5e-Compact. The flash sector utilities simplify the interface providing a means to develop interfaces that work across these platforms. For all new development you must use M5e Flash Addressing.

Flash Memory Sector Mapping

Sector	Access	Code	M5e/M5e-Compact Start Addr	M5e/M5e-Compact Size (bytes)
BootLoader	Read Only	0x01	0x000000	8 kB
Application	Read/Write	0x02	0x004000	224 kB
User Memory	Read/Write	0x03	0x03C000	16 kB
Hardware Info	Read Only	0x04	0x002000	8 kB

Accessing the Flash

The flash is accessed only through the boot loader program. Flash is not accessible while the application FW is running.

All accesses to flash are in terms of two-byte word addresses and word lengths. Thus, the **Write Flash Sector**, **Read Flash Sector**, **Erase Flash Sector**, and **Modify Flash Sector** commands all use the same argument types. The maximum amount of flash that can be written or modified at a single time is 125 words, and the maximum amount of flash that can be read at a single time is 124 words. This is a limitation of the serial interface and data packet sizes. Multiple data packets are used to read/write/modify a larger area of flash.

When using the **Erase Flash Sector** or **Write Flash Sector** commands, the correct password must be provided to complete the operation. This is done to protect against accidentally erasing or writing to the flash. See Boot Loader Commands.

Upgrading Application FW

The application FW is upgraded in flash. New versions of firmware are released in a or .sim binary file format for the M5e and M5e-Compact.The .sim binary file format is a compressed file format that stores the data in raw binary.

Direct Flash Addressing

The modules also provide a direct interface to the flash memory. This is a legacy interface and is not recommended for new development efforts.

M4e Direct Flash Addressing

This interface divides the flash into 11 different sectors of varying sizes. Only whole sectors of flash can be erased at a time. However, any amount of flash can be written provided that it is currently blank. This interface should no longer be used,

Table 5 shows the memory map for the flash chip. Notice that only sector 9 is set aside for user data and the other sectors are used by the application FW.

Flash memory map

Sector	Word Address	Size (kB)	Erase Password	Write Password	Comments
0	0x200000	32	--	--	Boot loader area
1	0x208000	32	0x08959121	0x02254410	Application FW
2	0x210000	32	0x08959121	0x02254410	Application FW
3	0x218000	32	0x08959121	0x02254410	Application FW
4	0x220000	32	0x08959121	0x02254410	Application FW
5	0x228000	32	0x08959121	0x02254410	Application FW
6	0x230000	32	0x08959121	0x02254410	Application FW
7	0x238000	16	0x08959121	0x02254410	Application FW
8	0x23C000	4	"	"	"
9	0x23D000	4	0x79138766	0x76346700	User Data
10	0x23E000	8	--	--	HW Calibration Data

Note

Addresses and sizes are shown in words (16-bits).The erase and write passwords are not provided for sectors 0 and 10. These sectors are considered to be the system area, and cannot be modified.

M5e Flash Addressing

The M5e and M5e-Compact memory map uses virtual addressing which contrasts to M4e which has hard-coded addresses. The M5e/M5e-Compact also has a User Data sector which is 16 kilobytes. The M5e is backward compatible to the M4e so that anyone familiar with M4e can still use the same command to write to the User Data sector. The software converts the M4e command and data to the M5e command. However, the M4e has a limit of 8 kilobytes of User memory instead of the 16 kilobytes for M5e/M5e-Compact. Therefore, it is preferable to use the **Write Flash Sector** command when writing user data to the M5e/M5e-Compact flash memory Write Flash Sector (0Dh).

Serial Port

The modules communicate to a host processor via the TTL logic level UART serial port, accessed on the 12-pin digital connector for M5e/M5e-Compact. See M5e/M5e-Compact Digital Connectors.

UART TTL Level TX

V-Low: Max 0.4 VDC
V-High: 2.9 to 3.3 VDC
8 mA max

UART TTL Level RX

V-Low: -0.3 to 0.8 VDC
V-High: 2 to 5.5 VDC
(Tied to 3.3 V through a pull-up resistor. Not harmful, but not recommended to drive the input above 3.3 V because then the input will be sourcing current to the 3.3 V supply.)

A level converter could be necessary to interface to other devices that use standard 12V RS-232. Only three pins are required for serial communication (TX, RX, and GND). Hardware handshaking is not supported. The serial port has an interrupt-driven FIFO that empties into a circular buffer.

The developer is responsible for ensuring that the host processor's receiver has the capability to receive up to 256 bytes of data at a time without overflowing.

General Purpose Inputs/Outputs (GPIO)

The Mercury Embedded modules have four TTL level signals, two input lines and two output lines, available on the 12-pin (M5e/M5e-Compact) digital connector. These can be controlled via the **Get User GPIO Inputs** and **Set User GPIO Outputs** commands.

Note

If you are Using a Multiplexer the available GPIO Outputs will be reduced to 1 or 0 depending on whether one or both are used as control lines as specified by Set Reader Configuration(9Ah).

For further information, see Get User GPIO Inputs (66h) and Set User GPIO Outputs (96h).

M5e and M5e-Compact GPIOs

The M5e has two 3.3/5V serial input sensor ports (GPIO inputs) and two output indicator ports (GPIO outputs) of up to 24 mA. The M5e-Compact has two 3.3/5V GPIO inputs and two GPIO outputs of up to 8 mA.

Default Settings

Since default settings may change across release and be different across platforms we recommend using the Get Application Commands to obtain default settings.

None of the configurable settings in the application FW are saved in non-volatile memory. Thus the system will always boot up in the same default state, regardless of how it was previously configured.

Overview of the Communication Protocol

The serial communication between a computer (host) and the reader is based on a synchronized command-response/master-slave mechanism. Whenever the host sends a message to the reader, it cannot send another message until after it receives a response. The reader never initiates a communication session; only the host initiates a communication session.

This protocol allows for each command to have its own timeout because some commands require more time to execute than others. The host manages retries, if necessary. The host keeps track of the state of the intended reader if it reissues a command.

Host-to-Reader Communication

Host-to-reader communication is packetized according to the following diagram. The reader can only accept one command at a time, and commands are executed serially, so the host waits for a reader-to-host response before issuing another host-to-reader command packet.

Header	Data Length	Command	Data	CRC-16 Checksum

Hdr	Len	Cmd		- - - - -		CRC Hi	CRC LO

1 byte	1 byte	1 byte	0 to *N* bytes	2 bytes

The fields are summarized in the following table.:

Field	Length	Description
Header (Hdr)	1 byte	Defines the start of the packet. Equal to 0xFF
[1]Data Length (Len)	1 byte	Defines the length, *N*, of the data field contained in the packet.
Command	1 byte	Specifies the command that the reader is to execute.
Data	*N* bytes (0 to 250)	Defines the binary data required by the reader for use with a command. This could, for example, represent transponder data to be written. The length, *N*, can vary between 0 and 250 bytes.
CRC-16 Checksum (CRC HI, CRC LO)	2 bytes	CRC-16 checksum (high order byte first). CRC polynomial is CCITT CRC-16, with a preload of 0xFFFF. This does not fully specify the operation of the CRC, see CCITT CRC-16 Calculation.

1.Minimum packet length is 5 bytes; the maximum packet length is 255 bytes.

Reader-to-Host Communication

The following diagram defines the format of the generic Response Packet sent from the reader to the host. The Response Packet is different in format from the Request Packet.

Header	**Data Length**	**Command**	Status Word	**Data**	**CRC-16 Checksum**
Hdr	**Len**	**Cmd**	Status Word	- - - - -	**CRC HI** **CRC LO**
1 byte	**1 byte**	**1 byte**	*2 bytes*	0 to *M* bytes	2 bytes

The fields are summarized in the following table.:

Field	Length	Description
Header (Hdr)	1 byte	Defines the start of the packet. Equal to 0xFF
[1]Data Length (Len)	1 byte	Defines the length, *M*, of the data field contained in the packet. Length can be 0 – 248 bytes
[2]Command	1 byte	OpCode of the last command received
[3]Status Word	2 bytes	Specifies the status of the last command, Successful = 0x0000, else it contains a fault code.
Data	*M* bytes (0 to 248)	Defines the binary data returned by the reader in response to a command. This could, for example, represent data read from a transponder. Data length, *M*, can be a minimum of 0 and a maximum of 248 bytes.
CRC-16 Checksum (CRC HI, CRC LO)	2 bytes	CRC-16 checksum (high order byte first). CRC polynomial is CCITT CRC-16, with a preload of 0xFFFF. This does not fully specify the operation of the CRC, see CCITT CRC-16 Calculation.

1.The minimum packet length is 7 bytes and the maximum packet length is 255 bytes.
2.Each host command receives a response from the reader. In the response packet, the Header, Data Length, Command, Data, and Checksum are functionally similar to the command packet.
3.The only difference is the addition of the Status Word field. The Status Word has two types of values. A Status Word value of 0 (Zero) means the command received was successful. Any other value represents a fault.

CCITT CRC-16 Calculation

The same CRC calculation is performed on all serial communications between the host and the reader. The CRC is calculated on the Data Length, Command, Status Word, and Data bytes. The header (SOH, 0xFF) is not included in the CRC.

A sample implementation of the CCITT CRC-16 algorithm is shown in this section. The **CRC_calcCrc8()** function is written to calculate the CRC one byte at a time, with the calculated value stored in crc_calc. The crc_calc value must be pre-loaded the first time the **CRC_calcCrc8()** function is called with 0xFFFF to initialize the calculated CRC. The final value of crc_calc is sent as the 16-bit CRC at the end of the message.

An example implementation of CRC calculation, taken from the Arbser source CrcUtils.c, is shown here:

```
/** @fn void CRC_calcCrc8(u16 *crcReg, u16 poly, u16 u8Data)
* @ Standard CRC calculation on an 8-bit piece of data. To make it
* CCITT-16, use poly=0x1021 and an initial crcReg=0xFFFF.
*
*  Note: This function allows one to call it repeatedly to continue
*        calculating a CRC. Thus, the first time it's called, it
*        should have an initial crcReg of 0xFFFF, after which it
*        can be called with its own result.
*
* @param *crcRegPointer to current CRC register.
* @param poly  Polynomial to apply.
* @param u8Datau8 data to perform CRC on.
* @return None.
*/

void CRC_calcCrc8(u16 *crcReg, u16 poly, u16 u8Data)

{
   u16 i;
   u16 xorFlag;
   u16 bit;
   u16 dcdBitMask = 0x80;


   for(i=0; i<8; i++)
   {
// Get the carry bit.  This determines if the polynomial should be
// xor'd with the CRC register.

         xorFlag = *crcReg & 0x8000;

         // Shift the bits over by one.
         *crcReg <<= 1;

         // Shift in the next bit in the data byte
         bit = ((u8Data & dcdBitMask) == dcdBitMask);
         *crcReg |= bit;

         // XOR the polynomial
         if(xorFlag)
         {
               *crcReg = *crcReg ^ poly;
         }

         // Shift over the dcd mask
         dcdBitMask >>= 1;
   }
}
```

Format for Microprocessor Reply to Host

There are three different types of replies that the microprocessor can make to the host as follows:

- Acknowledge that the command was properly processed (ACK)
- Return a fault code
- Provide data that is requested by the host

This section describes each of these three types.

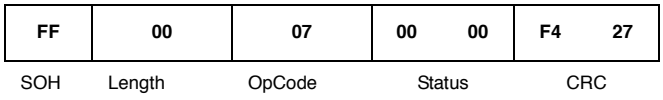
Unless otherwise specified, all commands return, as part of the Reply message, a status word with an ACK or a Fault. Those commands that return a Data Reply message are clearly shown.

Microprocessor ACK Message

Many of the commands require the microprocessor to perform a function, but do not require the microprocessor to send data back to the host. However, since the host cannot send a message until the microprocessor replies, an ACK is sent.

The ACK message contains no data. It returns the same OpCode that was sent originally to the microprocessor, sets the Status Word to 0x0000 (zero) and the Data Length to 0x00 (zero).

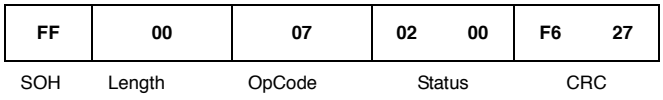
The following shows an example of an ACK message to an **Erase Flash** command.



The value in the OpCode field (0x07) is the same as the **Erase Flash** OpCode, 0x07.

Microprocessor Fault Reply Message

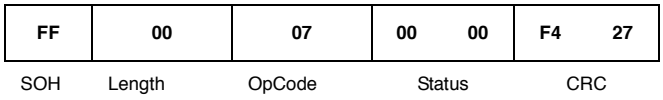
If a problem occurs during the execution of a command, the microprocessor returns a non-zero status value. Although this usually implies a fault or error, sometimes the non-zero status simply indicates a condition of the system. For instance, when executing a **Read Tag Single** command, if no tags are found, a status code of 0x0400 is returned. An example of a fault reply message is shown for the **Erase Flash** command.



A list of error codes is included in Appendix C: Error Messages. Refer to this list when encountering any non-zero status codes.

Microprocessor Data Reply Message

If the requested command requires that the microprocessor returns data, then the microprocessor creates a message similar to the Microprocessor ACK Message with the data length set to a non-zero value. Since this command does not require a data field, the length field is set to Zero.



Here is an example of a Reply message with Data Field Length not zero. This message happens to be a successful reply to **Read Tag Single** command.

FF	0A	21	00	00	C8	05	07	A8	00	84	C4	FF	9E	E0	F7	25
SOH	Length	OpCode	Status		Tag ID								Tag CRC		CRC	

Command Set

The following list defines the OpCodes that are used in the embedded modules firmware. As these products grow, more OpCodes will be added to enhance the functionality of the product. The timeout for the commands are in milliseconds. The maximum value for any user-configurable timeout is 65,535msec (0xFFFF), unless otherwise noted. The OpCodes are divided into five categories:

- 0x00 – 0x1F Boot Loader Commands
- 0x20 – 0x5F Application Tag Commands
- 0x60 – 0x8F Get Application Commands
- 0x90 – 0xBF Set Application Commands
- 0xC0 – 0xCF FCC Test Commands

Note

See Minimum Set of Serial Commands for the minimum startup command sequence required to configure the module and read tags.

Format for Microprocessor Reply to Host

There are three different types of replies that the microprocessor can make to the host as follows:

- Acknowledge that the command was properly processed (ACK)
- Return a fault code
- Provide data that is requested by the host

This section describes each of these three types.

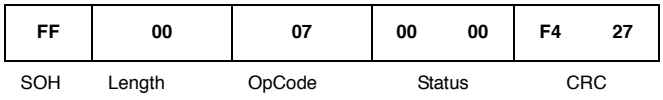
Unless otherwise specified, all commands return, as part of the Reply message, a status word with an ACK or a Fault. Those commands that return a Data Reply message are clearly shown.

Microprocessor ACK Message

Many of the commands require the microprocessor to perform a function, but do not require the microprocessor to send data back to the host. However, since the host cannot send a message until the microprocessor replies, an ACK is sent.

The ACK message contains no data. It returns the same OpCode that was sent originally to the microprocessor, sets the Status Word to 0x0000 (zero) and the Data Length to 0x00 (zero).

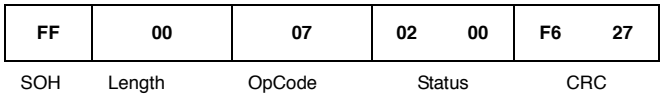
The following shows an example of an ACK message to an **Erase Flash** command.



The value in the OpCode field (0x07) is the same as the **Erase Flash** OpCode, 0x07.

Microprocessor Fault Reply Message

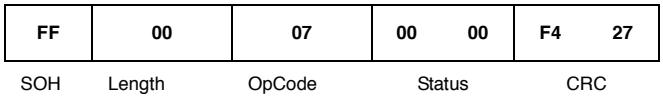
If a problem occurs during the execution of a command, the microprocessor returns a non-zero status value. Although this usually implies a fault or error, sometimes the non-zero status simply indicates a condition of the system. For instance, when executing a **Read Tag Single** command, if no tags are found, a status code of 0x0400 is returned. An example of a fault reply message is shown for the **Erase Flash** command.



A list of error codes is included in Appendix C: Error Messages. Refer to this list when encountering any non-zero status codes.

Microprocessor Data Reply Message

If the requested command requires that the microprocessor returns data, then the microprocessor creates a message similar to the Microprocessor ACK Message with the data length set to a non-zero value. Since this command does not require a data field, the length field is set to Zero.



Here is an example of a Reply message with Data Field Length not zero. This message happens to be a successful reply to **Read Tag Single** command.

FF	0A	21	00	00	C8	05	07	A8	00	84	C4	FF	9E	E0	F7	25
SOH	Length	OpCode	Status		Tag ID								Tag CRC		CRC	

Boot Loader Commands

The BootLoader is automatically started upon power up, and allows access to the on-board flash memory along with other commands. The program exits only when the **Boot Firmware** command is received. Once that occurs, the firmware image starts executing and sends back a reply to the **Boot Firmware** command. The BootLoader can also be started using command 0x09, **Start BootLoader**.

With the M5e/M5e-Compact, ThingMagic has created a hardware-neutral version of the bootloader. The reason for this change was that the M4e modules had an interface that was not easily portable to future modules. The M5e and M5e-Compact use an abstract version of these commands to provide easy inter operability between modules. The affected commands are:

- 0x01 – Write flash by address. Will be deprecated in future releases, replaced by 0x0D.
- 0x02 – Read flash memory. Will be deprecated to no longer accept 6 arguments.
- 0x06 – Set baud rate. Will be deprecated to only accept u32 baud rates.
- 0x0A – Modify flash. Will be deprecated in future releases, replaced by 0x0F.
- 0x0D – Write flash by sector.
- 0x0E – Get sector size.
- 0x0F – Modify flash by sector.

It is recommended to use the new interface wherever possible, since future products will no longer support the old ones as noted above. The M5e/M5e-Compact support most of the old interfaces (to ease transition for existing users) but support is not 100% guaranteed.

The following table shows which commands are supported by the Boot loader, the application, and the modules.

Boot Loader Commands

OpCode	Command Name	Bootloader	App Firmware
0x02	Read Flash (02h)	Y	N
0x03	Get Boot Loader/Firmware Version (03h)	Y	N
0x04	Boot Firmware (04h)	Y	N
0x06	Set Baud Rate (06h)	Y	Y
0x08	Verify Image CRC (08h)	Y	N
0x09	Start Bootloader (09h)	Y	N
0x0C	Get Current Program (0Ch)	Y	Y
0x0D	Write Flash Sector (0Dh)	Y	N
0x0E	Get Sector Size (0Eh)	Y	N
0x0F	Modify Flash Sector (0Fh)	Y	N

Read Flash (02h)

The **Read Flash** command reads the contents of flash from the specified address. Since the length of the Microprocessor reply packet is limited to 248 bytes, reading the application firmware data requires multiple read commands.

Note

Length is defined as the number of 16-bit words, and the maximum value of length is 124 words.

M5e/M5e-Compact Flash Read Example

There is an option where the M5e/M5e-Compact can read flash by sector number to decouple the memory map from the serial interface. From now on, all embedded products will use this interface.

FF	06	02	00 00 00 00	02	05	FA 59
SOH	Length	OpCode	Start Address	Sector	Num Bytes To Read	CRC

In this example, the sector number 02 indicates the application area. This command reads the first five data elements of the application, which is equivalent to the original command (using data length of 05). The sector codes can be found in Flash Memory Sector Mapping.

The reply to this command looks like this

FF	0A	02	00 00	01 23	45 67	89 AB	CD EF	01 23	BC ED
SOH	Length	OpCode	Status	[1]Word 1	[2]Word 2	Word 3	Word 4	Word 5	CRC

1. Word 1 contains the data located at address 0x208000.
2. Word 2 contains the data at address 0x208001, and so forth. (Remember that the Microprocessor data is word addressed.)

Error Status Codes

- FAULT_FLASH_ILLEGAL_SECTOR – 303h
- FAULT_MSG_INVALID_PARAMETER_VALUE - 105h

Get Boot Loader/Firmware Version (03h)

The **Get Boot Loader** command returns the Boot Loader, Hardware, and Application version numbers. The Boot Loader, Hardware, and Application FW version numbers are stored in flash. The Boot Loader and Hardware version numbers are each 32-bit numbers. The application has a 96-bit version code. The command to retrieve firmware version is shown in the following table.

FF	00	03	1D 0C
SOH	Length	OpCode	CRC

M5e-Compact and M5e Command Responses

A sample response for the M5e-Compact to the command is shown in the following table:

FF	14	03	00 00	07 09 17 00	01 00 00 01	20 07 10 12	09 05 12 00
SOH	Length	OpCode	Status	BootLoader Ver	Hardware Ver	Firmware Date	Firmware Version

00 00 00 10	6B CC
Supported Protocols	CRC

The following information is embedded in the reply to the command:

- The boot loader version is 07.09.17.00. This number is in hex format.
- HW Version is 01000001. For a definition, see Returned HW Versions.
- The application firmware was compiled on 2007-October-12.
- The application firmware version is 09.05.12.00. This number is in hex format.
- The protocol supported by this firmware – Gen2 and ISO 18000-6C.

A sample response for the M5e to the command is shown in the following table:

FF	14	03	00 00	07 09 06 00	00 00 00 03	20 07 10 04	09 05 12 00
SOH	Length	OpCode	Status	BootLoader Ver	Hardware Ver	Firmware Date	Firmware Version

00 00 00 10	AD 6E
Supported Protocols	CRC

Returned Hardware Version Table

The following table provides a definition of each HW version returned by the **Get Boot Loader/Firmware Version** command.

Returned HW Versions

Module	Defined	Version	Description
M4e	HW_VERID_0_5W	0xFFFFFFFF	0.5 W
	HW_VERID_1_0W_NON_ROHS	0x01010000	original non-RoHS compliant curve (downward slope) and power range 20-30 dBm
	HW_VERID_1_0W_ROHS_EXT	0x03020000	RoHS compliant curve (upward slope) and power range 5-30 dBm
	HW_VERID_1_0W_NON_ROHS_EXT	0x03010000	original non-RoHS compliant curve (upward slope) and power range 5-30 dBm
M5e	HW_VERID_1_0W	0x00000001	1W, small form-factor, Impinj Indy1000 ASIC; Pre-production hardware, contact support@thingmagic.com.
		0x00000002	Noise improvements
		0x00000003	Adds LBT and DRM
M5e EU	HW_VERSION_M5E_EU_REV_1	0x02000001	Provides full 1 Watt power in EU region.
M5e-Compact	HW_VERID_0_2W_COMPACT	0x01000001	0.2 W, compact size, Impinj Indy1000 ASIC; Supports up to firmware version 1.0.22.
		0x01000003	Supports reduced power consumption

If the **Get Boot Loader/Firmware Version** command returns a different value than those listed in the table, you should contact support@thingmagic.com.

Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h

Boot Firmware (04h)

The **Boot Firmware** command tells the boot loader to run the current firmware image stored in flash in the following sequence:

1. The boot loader verifies the checksum of the application image before it is run.
2. If the image is invalid, a fault code is returned to the host.
3. If the application firmware is started successfully, it sends the response to the **Boot Firmware** command.

   The response is identical to the response to a **Get Version** command, except that the OpCode is 0x04 instead of 0x03.

Note

The **maximum time** required to boot the application firmware **is 650ms.** There will be release to release variation in actual boot time but it will always be less than the maximum.

Error Status Codes

- FAULT_BL_INVALID_IMAGE_CRC – 200h

Set Baud Rate (06h)

The **Set Baud Rate** command has a default baud rate of 9600 bps. Since the code method of specifying the baud rate is processor dependent, a new interface was created to make the modules easily interchangeable.

The following table shows the hexadecimal equivalent for each baud rate:

Baud Rate (decimal)	Baud Rate (hex)
9600	0x00002580
19200	0x00004B00
38400	0x00009600
57600	0x0000E100
115200	0x0001C200
230400	0x00038400

460800[1]	0x00070800
921600[1]	0x000E1000

Note: 1 - 460800 and 921600 cannot be used in bootloader mode, only application mode.

C A U T I O N !

When using the 921600 baud rate you must pass an extra byte of data (0x00) at the end of each message, after the CRC. This is due to a timing issue. If the extra byte is not passed the message will not be processed until the another byte of data is received.

In the following example, the baud rate is specified as a 32-bit value. This example sets the baud rate to 115200:

FF	04	06	00	01	C2	00	A4	60
SOH	Length	OpCode	Baud Rate				CRC	

The response to baud rate change is sent at the baud rate that the **Set Baud Rate** command was transmitted. Once the baud rate has changed, the new rate is in effect until the baud rate is changed by power cycling the reader.

Note

The baud rate reverts back to 9600 bps at power-up.

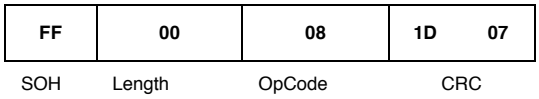
Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h
- FAULT_INVALID_BAUD_RATE - 10Ah

Verify Image CRC (08h)

After uploading a new application firmware image, the application CRC can be checked with the **Verify Image CRC** command. The application CRC is already embedded in the firmware image, so this command calculates the firmware's CRC in flash and compares it to the pre-stored value. It returns a fault code if the application firmware fails the CRC

checks. A failed CRC means that the application cannot run, and needs to be downloaded again.



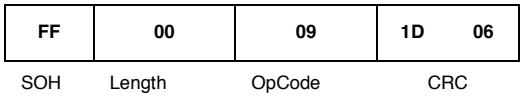
The boot loader runs this command automatically before loading the application. If the CRC check fails, the boot loader does not run the application to prevent corrupted code from executing on the Microprocessor.

Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h
- FAULT_BL_INVALID_IMAGE_CRC – 200h

Start Bootloader (09h)

The **Start Bootloader** command shuts off the analog board and starts the boot loader while inside the application. This is necessary to perform an upgrade of the firmware while the system is running.



Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h

Get Current Program (0Ch)

The **Get Current Program** command returns a code for the current program being executed in the module. These codes are defined as follows:

Code	Program
0x11	M5e/M5e-C Bootloader
0x12	M5e/M5e-C Application

To get the current program command, send the following to the module:

FF	00	0C	1D 03
SOH	Length	OpCode	CRC

The module responds as follows, indicating that it is a M5e/M5e-Compact application program:

FF	01	0C	00 00	12	63 43
SOH	Length	OpCode	Status	Program	CRC

Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h

Write Flash Sector (0Dh)

The **Write Flash Sector** command is the same as the old **Write Flash** (01h) except it has an extra flash sector argument. This allows future Mercury Embedded products to work with the same command set.

FF	0F	0D	02 25 44 10	00 00 00 00	02	12 34 56 78 90 12	73 4C
SOH	Length	OpCode	Password	Start Address	Sector	Data To Write	CRC

Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h
- FAULT_FLASH_ILLEGAL_SECTOR – 303h
- FAULT_FLASH_BAD_WRITE_PASSWORD – 301h

Get Sector Size (0Eh)

The size of a flash sector can be retrieved from the module using the **Get Sector Size** command. Since different products may have different flash sector sizes, this command is useful for ensuring that the module has enough memory to store the desired data. This example receives the sector size for the application area:

FF	01	OE	02	D1 BF
SOH	Length	OpCode	Sector	CRC

The response to this command on M5e is shown in the following example:

FF	04	OE	00 00	00 03 40 00	88 54
SOH	Length	OpCode	Status	Size of Sector	CRC

The size of the sector is returned in bytes. For M5e, sector 2 (application) is 212992 bytes.

Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h
- FAULT_FLASH_ILLEGAL_SECTOR – 303h

Modify Flash Sector (0Fh)

The **Modify Flash Sector** command is the new version of the original **Modify Flash** (0Ah) command. The following example writes 6 bytes to Sector 03.

FF	0F	0F	79 13 87 66	00 00 00 00	03	12 34 56 78 90 12	4C FA
SOH	Length	OpCode	Password	Start Address	Sector	Data To Write	CRC

Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h
- FAULT_FLASH_ILLEGAL_SECTOR – 303h
- FAULT_FLASH_BAD_ERASE_PASSWORD – 300h

Application Tag Commands

The application commands are used to interact with RFID tags in the field. These commands can have slightly different behavior based upon the current protocol selected in the system.

Applications Commands

OpCode	Command Name	Bootloader	App Firmware
0x21	Read Tag Single (21h)	N	Y
0x22	Read Tag Multiple (22h)	N	Y
0x23	Write Tag EPC (23h)	N	Y
0x24	Write Tag Data (24h)	N	Y
0x25	Lock Tag (25h)	N	Y
0x26	Kill Tag (26h)	N	Y
0x28	Read Tag Data (28h)	N	Y
0x29	Get Tag Buffer (29h)	N	Y
0x2A	Clear Tag Buffer (2Ah)	N	Y
0x2D	Gen2 Tag Specific (2Dh)	N	Y
0x2E	BlockErase (2Eh)	N	Y

Tag Singulation/Select Functionality

Many of the Gen2 tag commands now support the ability to singulate a specific tag or inventory only tags matching a defined criteria, i.e. matching on values in the EPC, TID and User Memory banks.

Select Algorithm and Parameters

The algorithm used to perform a Gen2 Select and its impact on subsequent Gen2 Query operations on a population of tags is determined by several user defined settings which either correspond directly to or are used to determine the various Gen2 Select and Query

parameters as defined by the EPCGlobal Gen2 v1.2 Specification. The current user controlled options are:

Gen2 Session (User Controlled)

This setting determines which tag inventory flag is altered when a tag responds to a Gen2 Query (each flag has a unique persistence profile, as defined by the Gen2 Specification). The value of Gen2 Session and Gen2 Target is set using the Set Protocol Configuration (9Bh) command.

Select Invert (User Controlled)

This setting is used to determine the Gen2 Action value sent in the Gen2 Select command. The value of Select Invert is defined using bit 3 of the Select Option field in the Tag Singulation Fields. The supported values are:

- **0** (False) - Tags which match the Select criteria are to respond. Set Gen2 Action=0.
- **1** (True) - Tags which DO NOT match the Select criteria are to respond. Set Gen2 Action=4.

The settings specified for Gen2 Session and Select Invert determine the settings within the Select command which is sent before the corresponding Gen2 Query and the settings in the subsequent Gen2 Query(s). These implicit settings used by the Gen2 Select are:

Gen2 Action (defined by Select Invert)

This parameter can be one of 8 values, as defined by the Gen2 specification, which determine which of 4 possible flag actions (assert, de-assert, negate, or leave alone) will be done if the criteria matches and, similarly, which of the 4 possible flag actions will be done if the criteria does not match. Only two values are currently used:

- **0** - Assert (or put in state “A” for inventory flags) the target flag if there IS a match; de-assert (or put in state “B” for inventory flags) the target flag if there IS NOT a match.
- **4** - Assert (or put in state “A” for inventory flags) the target flag if there IS NOT a match; de-assert (or put in state “B” for inventory flags) the target flag if there IS a match

Gen2 Target (User Controlled)

This setting determines which inventory flag or SL flag is going to have its state determined by the matching algorithm. Currently always set to ‘4’, indicating the Gen2 Select command modifies a tag’s SL flag.

The following table defines the currently supported User Settings and the resulting behavior of the Gen2 Select and Gen2 Query:

Gen2 Select and Query Behavior

User Settings	Select Behavior	Query Behavior	Comments
• Gen2 Session = 0,1,2,3 • Invert = 0 • Gen2 Target = A, B, A<>B, B<>A	Gen2 Select Settings: • Target = 4 • Action = 0 If tags match the criteria, put their SL flag in the ‘Assert’ state; and SL of non-matching tags into the ‘De-assert’ state.	Gen2 Query Settings: • SEL = 3 • Target = [User Defined] • Session = [User Defined] Ask tags to respond if their SL flag is in the ‘Assert’ state and their session appropriate inventory flag is in the user defined state (default is ‘A’). Once a tag responds, it puts the inventory flag in the opposite state (default is ‘B’), preventing further matches until the inventory flag’s Flag Persistence Rules changes it back.	Tag state persistence before a Query is based on SL flag persistence; Tag state persistence after a Query is based on inventory flag corresponding to the Session used.
• Gen2 Session = 0,1,2,3 • Invert = 1 • Gen2 Target = A, B, A<>B, B<>A	Gen2 Select Settings: • Target = 4 • Action = 4 If tags match the criteria, put their SL flag in the ‘De-assert’ state; and SL of non-matching tages into the ‘Assert’ state.		

Select Process

The following defines how the Select process works when attempting to select tags that match a defined criteria:

1. The Reader issues a Select containing the desired tag memory values and instructions for the tag to assert if its contents matches that specified in the request (and, conversely, de-assert the SL flag if it does not match) as defined by the Tag Singulation Fields. The de-assert will generally have no effect because the de-asserted state is the default, but is helpful if tags still have their SL flag asserted from a previous Select.
   – Tags that are selected, at this point, are not selected within a specific Session. The persistence of their state depends entirely on that of the SL flag.
2. A Query is issued which specifies the flag settings which must match before a tag will respond:

- The SL flag must be asserted
- The Session flag (Target setting) for session 0, 1, 2, or 3 (specific Session value is in the Query and depends on the reader's Session setting) must be in the user defined state (the default value is 'A').

3. Matching tags will respond to the Query, but after responding, will change their own state in the following way:

- The Inventory flag corresponding to the Session specified in the Query will be changed to the opposite state (A->B or B->A).
- The SL flag will remain asserted (per its ongoing Flag Persistence Rules)

4. If subsequent indentical Queries are issued (identical to the first), this tag will remain silent until the Flag Persistence Rules for the inventory flag that was put into the 'B' state cause the flag to fall back into the 'A' state or vice-versa. At that point, the tag will respond again (assuming that the persistence rules of the SL flag are still keeping it in the 'assert' state). Unless the Target Value was set to search for A then B or B then A, in which case tags will be re-inventoried when the reader changes the flag value its searching for instead of waiting for the tag to put its tag back into the opposite state.

Note

A search with **Invert=1** specified will perform the same steps except in step 1 the tag will "**de-assert** if its contents matches that specified in the request (and, conversely, **assert** the SL flag if it does not match) as defined by the Tag Singulation Fields."

Note

A Select will be performed once per antenna when Select is specified during a Read Tag Multiple (22h) with multiple antennas configured.

Flag Persistence Rules

Session 0

- Keeps state as long as tag is energized
- Returns to default state as soon as the tag is no longer energized

Its state will often get reset during the course of executing a single command, for example, between inventory rounds, or when there is a frequency hop, or when a new antenna is selected during a Read Tag Multiple search. the result is that when the session is set to '0', all tags will respond to every appropriate Query, considerably lengthening the time to inventory a large population of tags. Session 0 is typically used for operations where a single tag is expected to be in the field, for example, printers.

Session 1

- Keeps its state between 0.5 and 5 seconds, regardless of whether the tag is energized or not.

The intent is that when the session is set to 1 the tag will respond to an appropriate Query immediately, and then respond periodically if the Query is repeated by the reader. This allows a larger population of tags to be reliable read. Session 1 is typically used in applications where a large population of tags is being continuously inventoried allowing for it to be determined when tags enter and leave the field.

Session 2, 3 and SL Flags

- Keeps state as long as the tag is energized.
- Keeps its state for at least 2 seconds after the tag is no longer energized and will refresh its state if the tag is re-energized during that period.

The intent is that when the session is set to '2' or '3' a tag will only respond once to an appropriate Query, then remain silent as the Query is repeated by the reader to elicit responses from other tags. Typically used in operations when you are performing an action on tag or population of tags which you do not want/need repeated.

Operations supporting Tag Singulation/Select

The commands currently supporting tag singulation through Select are:

- Read Tag Single (21h)
- Read Tag Multiple (22h)
- Write Tag EPC (23h)
- Write Tag Data (24h)
- Lock Tag (25h)
- Kill Tag (26h)
- Read Tag Data (28h)
- Gen2 Tag Specific (2Dh)
- BlockWrite (2Dh)
- BlockPermaLock (2Eh)
- BlockErase (2Eh)

The addition of this functionality has added several (some conditional) fields to these commands. The deprecated version of these commands are listed in Appendix D:

Release Version 1.0.34. The command information in this section has been updated to *only* include the new format.

The following fields have been added to all the specified commands. Please check each command for exact order, and any exceptions, as they may not all correspond with the order below.

Tag Singulation Fields

Field		Values	Description
Select Options	[1]Select Contents (Bits 0,1, 2)	0x00	Select functionality is disabled. First tag found will be the tag operated on. No other Tag Singulation Fields should be specified. Option field must **always** be specified. Note: When Select is disabled commands do not support an access password. Use Select Option=0x05 to send a password without Select.
		0x01	Select on the value of the EPC. Requires all fields except the *Select Address* field.
		0x02	Select on contents of TID memory bank (Gen2 bank 0x02). Requires all fields.
		0x03	Select on contents of User Memory memory bank (Gen2 bank 0x03). Requires all fields.
		0x04	Select on contents of the EPC memory bank (Gen2 bank 0x01). Requires all fields.
		0x05	Use this option when you need to specify an access password to operation on locked data but don't want to perform a Select. When this option is used do not pass any Select Criteria.
	[1]Select Invert (Bit 3)	0x08	Sets Invert Flag. This results in tags NOT matching the specified Tag Singulation Fields will be returned, as defined in Select Algorithm and Parameters.
	[1]Extended Select Data (Bit 5)	0x20	Changes *Select Data Length* to 2 bytes, allowing *Select Data* to be greater than 255 bits.

Field	Values	Description
The Select Options field is typically followed by command specific fields. After the command specific fields the following Tag Singulation fields should be specified as appropriate for the Select Contents specified.		
Select Address	4 bytes	Contains the offset, in bits, within the memory bank, specified by the *Option* value, at which the comparison is to start. NOTE: specifying *Option=0x04* and *Select Address=0x20* is the equivalent, for Gen2 v1 tags, of specifying *Option=0x01*, both specify a comparison against the tag EPC ID data. Note: Addresses are always zero-based. Specifying 0x00 indicates starting at the first address location.
Select Data Length	1 byte (*2 bytes if Extended Data enabled*)	Contains the length of the data (*Select Data*) to be compared, in bits, to the EPC when *Option*=0x01, or to the data beginning at *Select Address* for other options.
Select Data	M bytes	Contains the data to be compared against the specified tag data (memory bank and address, or EPC as specified by the *Option* value) The bit values used start at address 0. So if *Select Data Length = 2*, i.e. matching 2 bits, then the bits used for comparision will be the 2 most significant bits of the *Select Data* value. Examples: Select Data = 0x00 the bits to match will be 0, 0 Select Data = 0x8F the bits to match will be 1, 0 This is independent of the *Select Data Address* field.
Note: 1- The Select Options field contains multiple sub fields which must be combined into a single Select Options value. This means the final Select Options value is a result of Select Invert + Select Contents.		

Example:

The following EPC IDs (first 3 bits) are in the field:

0xAAAA (101)
0xCCCC (110)
0x4444 (010)
0x3000 (001)

Select Option = 0x04 (EPC Mem Bank)
Select Data Length = 0x01 (1 bit)
Select Data = 0x80
Select Data Address = 0x00000022 (third bit in the EPC ID)

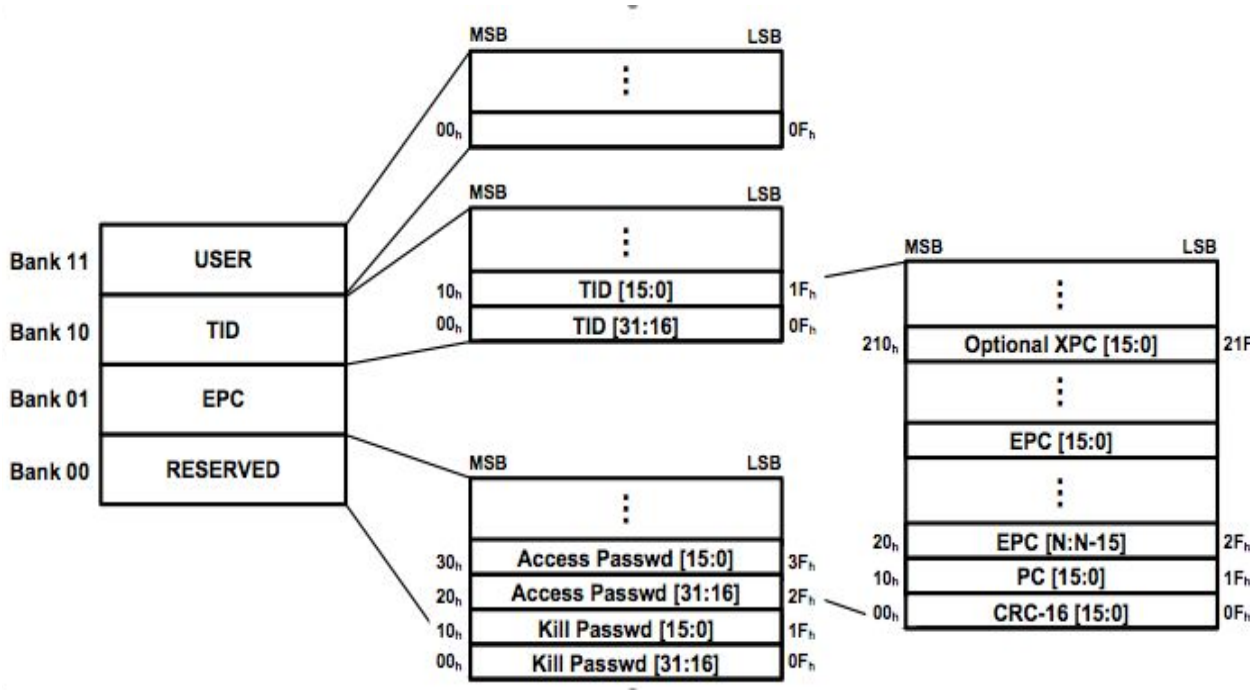
In this case the third bit of the EPC ID is matched against the first bit of the *Select Data* value, 1. This would result in the following IDs being returned:

0xAAAA
0x3000

Gen2 Memory Map

When performing a Tag Singulation/Select most of the criteria specifies values of data in certain locations in a Gen2 tag's memory map. The following is a logical view of the Gen2 memory map from the Generation2 Protocol v1.2 that can be used for reference when trying to determine the memory address you are trying to match on:

Gen2 Tag Memory Map



Note

The address values specified in the memory map are hexidecimal, zero-based bit offset within each memory bank (i.e. the PC section of the EPC memory bank starts at bit 0x10, decimal 16, and runs to bit 0x1F, decimal 31). It is important to note the units used in various command fields for address locations. In some cases the address is specified in words (16 bit chunks), sometimes bytes (8 bit chunks) and sometimes bits.

Read Tag Single (21h)

For deprecated version (including non-Gen2 protocol specific syntax) of this command see Appendix D: Read Tag Single (21h)

The **Read Tag Single** command will search for a tag for the specified timeout or a single tag is found, whichever comes first. The search criteria is specified using the Tag Singulation Fields. If Option=0x00 is specified it will return with the first tag it finds, otherwise it will only return Success and the found EPC if a tag matching the specified criteria is found. If no tag is read, a fault code is returned.

In addition to the Tag Singulation Fields the basic **Read Tag Single** command takes a 16-bit timeout value in milliseconds. The command will return after a tag is found or the timeout expires, whichever happens first.

The basic syntax which returns only the tag EPC is defined in Get Tag EPC. With additional Option bits set Read Tag Single can also return Tag Read Meta Data using the syntax in Get Tag EPC and Meta Data.

Note

Read Tag Single will always use Gen2 Q=0 when Get Protocol Configuration (6Bh) Q Value set to Dynamic Q. For use with large tag populations a Static Q appropriate for the population should be used to avoid collisions.

Get Tag EPC

The following example shows a search requesting a tag matching the following criteria for a max timeout of 1000 ms. This example uses the Tag Singulation/Select Functionality with Option=0x03, indicating Tag Selection based on the contents of User Memory, specifically:

Memory Bank = User Memory.

Starting Address = bit 32

Select Data = 0x1234

FF	0A	21	03 E8	03	00 00 00 20	10	12	34	E5 AC
SOH	Length	OpCode	Timeout (ms)	Option	Select Address	Select Data Length	Select Data		CRC

If Option=0x00 or 0x01 were used then the unused Tag Singulation Fields must be removed from the request.

The response to this command varies depending upon the number of bits in the tag EPC of the tag found. The general response format is shown here:

FF	M+3	21	00 00	03	M bytes	?? ??	?? ??
SOH	Length	OpCode	Status	Option	EPC	TagCRC	CRC

Get Tag EPC and Meta Data

In addition to getting the tag EPC value returned you can also get Tag Read Meta Data for the found tag. This version of Read Tag Single requires bit 4 of the Option flag to be set and takes an additional Metadata Flags field which defines what metadata will be returned The following table lists the supported values for these fields.

Read Tag Single Get EPC and Metadata Request Fields

Field	Value	Description
Option	Bit 4=0 (0x0X)	No Metadata flags are specified and Meta Data will not be returned. This is the Get Tag EPC syntax. The lower bits (X) are specified as defined by Tag Singulation/Select Functionality.
	Bit 4=1 (0x1X)	Indicates that Metadata flags are to follow and the corresponding Metadata shall be returned with the tag EPC. The lower bits (X) are specified as defined by Tag Singulation/Select Functionality.
Metadata Flags (to specify more than one OR the values together)	0x0000	When no flags are set no meta data will be returned, only the tag EPC (including the tag CRC)
	0x0001	When bit 0 is set the Read Count will be returned
	0x0002	When bit 1 is set the LQI/RSSI will be returned
	0x0004	When bit 2 is set the Antenna ID will be returned
	0x0008	When bit 3 is set the Frequency will be returned
	0x0010	When bit 4 is set the Timestamp will be returned
	0x0020	When bit 5 is set the RFU (T*hingMagic Only*) will be returned
	0x0040	When bit 6 is set the Protocol ID will be returned.
	0x0080	When bit 7 is set Tag Data information will be returned. (0x0000 is always returned for Read Tag Single)
These fields are followed by the Tag Singulation/Select Functionality, used the same as defined in the Get Tag EPC syntax, if necessary.		

A response can contain the following information:

Read Tag Single Get EPC and Metadata Response Fields

Field	Length	Value
SOH	1 byte	0xFF
Length	1 byte	Based on data returned
OpCode	1 byte	0x21
Status	2 bytes	Status of command
Options	1 byte	As sent in request
Metadata Flags	2 bytes	Metadata contained in response
Read Count[1]	1 byte	Tag EPC/Antenna Read Count
RSSI[1]	1 byte	Return Signal Strength Indicator
Antenna ID[1]	1 byte	Antenna ID, 4 MSBs for TX and 4 LSBs for RX
Frequency[1]	3 bytes	Frequency in kHz
Timestamp[1]	4 bytes	RTC Timestamp
RFU[1]	2 bytes	Reserved for Future Use - ThingMagic Only
Protocol ID	1 byte	Protocol ID of tag (always 0x05 for M5e/M5e-C)
Tag Data Length	2 bytes	Size of tag data to follow. Always 0x0000 for Read Tag Single
EPC ID	N bytes	Tag EPC.
Tag CRC	2 bytes	Tag EPC CRC
CRC	2 bytes	Message CRC

1 - Conditionally returned depending on the Metadata Fields specified in the request.

Examples

An example command requesting AntennaID and Timestamp

```
Metadata Flags = 0x0004 OR 0x0010 = 0x0014
```

with no tag singulation criteria, just return the first tag found, is as follows:

FF	05	21	01	E8	10	00	14	2F 6D
SOH	Length	OpCode	Timeout		Options	Metadata Flags		CRC

Here is an example response to the example request specified above. The response contains the tag EPC info of the found tag and the requested tag read metadata: AntennaID and Timestamp:

FF	16	21	00 00	10	00	14
SOH	Length	OpCode	Status	Options	Metadata Flags	

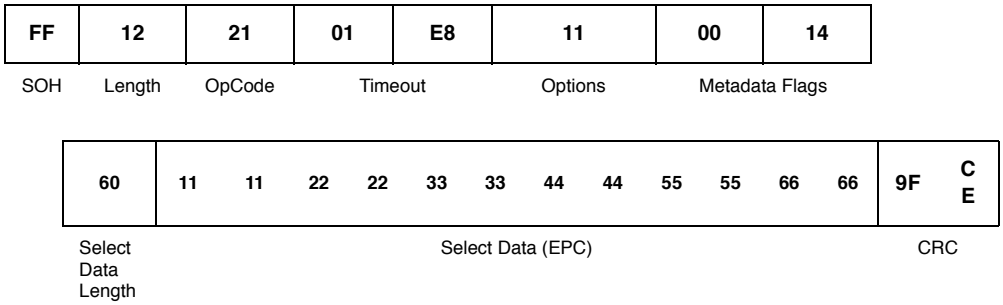
11	00	BB	5F	04	01	23	45	67	89	AB	CD	EF	01	23	45	67	E6	C8
Ant ID	Timestamp				Tag EPC												Tag CRC	

37	C4
CRC	

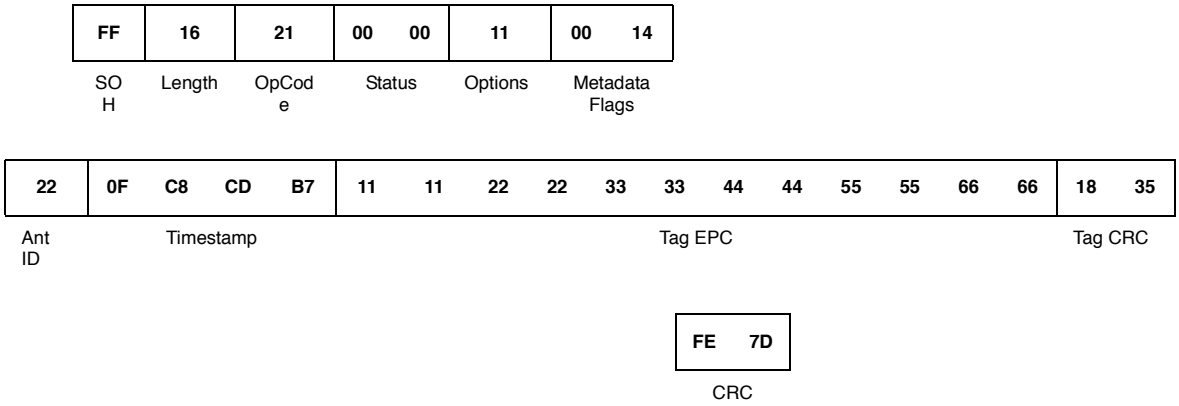
Here is another example request and response showing the use of Tag Singulation/Select Functionality and getting the tag metadata for the specified tag. Note the Options Field includes the Select Options and bit 4 is set indicating the Meta Data flags follow.

This command requests the same tag read metadata as the previous example (AntennaID and Timestamp) except now it is selecting a tag with a specific EPC value (EPC=0x111122223333444455556666), which requires adding the appropriate Tag Singulation Fields after the Read Tag Single Get EPC and Metadata Request Fields along with

updating the Option field to set the appropriate flag for the tag singulation based on EPC value



The response contains the requested metadata and the tag EPC matching the requested tag EPC:



Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h
- FAULT_MSG_INVALID_PARAMETER_VALUE - 105h
- FAULT_NO_PROTOCOL_DEFINED – 401h
- FAULT_AFE_NOT_ON – 405h
- FAULT_NO_TAGS_FOUND – 400h
- FAULT_ANTENNA_NOT_CONNECTED – 503h
- FAULT_TEMPERATURE_EXCEED_LIMITS – 504h
- FAULT_HIGH_RETURN_LOSS – 505h

Read Tag Multiple (22h)

For deprecated version of this command see Appendix D: Read Tag ID Multiple (22h)

The **Read Tag Multiple** command supports several different levels of functionality. In addition to performing a Basic Search Operation for all tags in the field it can also perform advanced searching and perform operations on the tags found. The different syntax for Read Tag Multiple are defined as follows:

- Basic Tag Inventory - Searches for and returns all tags in the field.
- Tag Inventory with Select - Searches for and returns all tags in the field meeting Select criteria as defined by Tag Singulation Fields specified.
- Tag Inventory With Embedded Operations - Allows for operations (Write Tag Data, Lock Tag, Kill Tag, Read Tag Data) to be performed on each tag inventoried.

Note

A Read Tag Multiple command will return early if it fills up the Tag Buffer before the timeout has expired. This will not result in an error. If a Read Tag Buffer command is issued with an **already full** Tag Buffer an error will be returned.

Basic Tag Inventory

The **Read Tag Multiple** command performs a search for the specified period of time then returns the number of tags that have been found. Afterwards, multiple **Get Tag Buffer** commands can be sent to receive the found tag EPCs along with tag read metadata, including the antenna the tag was read on. The command allows the user to specify the method to use when multiple antennas are configured and connected along with indicating the command contains embedded commands:

Read Tag Multiple Search Flags

[2]Flag Value		Description
Antenna Usage (bits 0, 1, 2)	0x0000	Use single antenna as configured by the most recent Set Antenna command.
	[1]0x0001	Automatically search on both monostatic antennas, **starting with Antenna 1**. The search cycles through antennas moving to the next antenna when no more tags are found on the current antenna. It stops when the search timeout expires.
	[1]0x0002	Automatically search on both monostatic antennas, **starting with Antenna 2**. The search cycles through antennas moving to the next antenna when no more tags are found on the current antenna. It stops when the search timeout expires.
	[1]0x0003	Automatically searches on all configured logical antennas, using the Search Order defined using Set Multi-Antenna Search Configuration. The search cycles through antennas, in the order specified, moving to the next antenna when no more tags are found on the current antenna. It stops when the search timeout expires.
Embedded Command (bit 3)	0x0004	An **embedded command** is specified in the request and will be executed on each inventoried tag. Note: This bit should only be set when using the Tag Inventory With Embedded Operations syntax.

Note: 1- Only one of these flags should be set since both Antenna 1 and 2 cannot be the starting antenna.

2 -Multiple Flags can be set (perform a binary OR) to specify different behaviors. ex: Search Flags = 0x0006 indicates a multiple antenna search starting on antenna2 is to be performed and the embedded command specified is to be executed on each tag.

Note: When performing multi-antenna searches and Tag Singulation/Select Functionality is used, the Select operation will be performed once per antenna. More specifically, during a single Tag Inventory with Select operation a Select will be issued at the start of searching on each antenna in use. If the search comes back around to an antenna during the same command the Select is not issued again.

Examples

For example, the syntax for a Read Tag Multiple with automatic multi-antenna search starting with antenna1 is:

FF	04	22	00 01	03 E8	3F 8E
SOH	Length	OpCode	Antennas Flag	Timeout (ms)	CRC

The response format for both is the following:

FF	01	22	00 00	02	46 BA
SOH	Length	OpCode	Status	# Tag IDs Found	CRC

Tag Inventory with Select

If you want to inventory only tags meeting a specific criteria this syntax should be used. The search criteria is specified using the Tag Singulation Fields. If Option=0x00 is specified it will perform the same search as the Basic Tag Inventory syntax. Otherwise, it will return the number of tags found matching the specified criteria. The tag EPCs and Meta Data will be available in the Tag Buffer. If no tags are found, a fault code is returned. The required fields are as follows:

Read Tag Multiple with Select Fields

Field	Value	Description
Select Options	[1 byte]	The Options value of the Tag Singulation Fields
Search Flags	[2 bytes]	Read Tag Multiple Search Flags indicating antenna usage. Bit 3 must be 0, no embedded commands.
Timeout	[2 bytes]	Indicates how long the command should spend searching.
Access Password	[4 bytes]	The Access Password is only used with Tag Inventory With Embedded Operations for the embedded command. With this syntax it should be specified as 0x00000000. Note: If Select Options=0x00 this field should be omitted.
Tag Singulation Fields		The remaining, appropriate fields depending on the value of Select Options.

Examples

Here is an example request and response showing the use of Tag Singulation/Select Functionality to inventory tags which meet a specific criteria:

This command will inventory all tags with an EPC value ending in 0x66, which requires adding the appropriate Tag Singulation Fields to the Basic Tag Inventory syntax

FF	0F	22	04	00 00	03 E8
SOH	Length	OpCode	Options (EPC Mem)	Search Flags	Timeout

00 00 00 00	00 00 00 78	08	66	D E C0
Access Password	Select Address (bits)	Select Data Length (bits)	Select Data	CRC

Note

The Select Options field of the Tag Singulation Fields in the request is specified at the beginning of the command followed by the Search Flags, Timeout, Access Password then the rest of the Tag Singulation Fields. This is different than the typical format for Select fields. Also, this syntax always requires an Access Password be specified. Since only Reserved Memory can be read locked and Reserved Memory cannot be used for singulation the Access Password must be 0x00000000

The response contains the number of tags found matching the Select criteria specified. Use Get Tag Buffer (29h) to access the tag EPCs and Tag Read Meta Data:

FF	04	22	00 00	04	00 00	02	B7 6E
SOH	Length	OpCode	Status	Options (EPC Mem)	Search Flags	Tag Found	CRC

Tag Inventory With Embedded Operations

In addition to inventorying tags, Read Tag Multiple can be used to perform an operation on each tag in a population of tags. Starting with the Tag Inventory with Select syntax to define the population of tags the operation is to be performed on, the Search Flag bit 3 (0x0004) can be set to indicate embedded commands are to be performed on the inventoried commands. The required fields are as follows:

Read Tag Multiple Embedded Command Fields

Field	Value	Description
Select Options	[1 byte]	The Options value of the Tag Singulation Fields
Search Flags	[2 bytes]	Bit 3 of the Read Tag Multiple Search Flags must be set indicating this request contains embedded command(s).
Timeout	[2 bytes]	Indicates how long the command should spend searching AND performing the embedded command. It may be desirable to specify a longer timeout if a large number of tags are likely to get the embedded command executed on them.
Access Password	[4 bytes]	The Access Password of the tags expected to be inventoried for the embedded command, if they are locked. If the tags are not locked specify 0x00000000. Note: If operating on locked tags, only tags which meet the Select criteria and matching passwords will get a successful execution of the embedded command. Note: If Select Options=0x00 this field should be omitted.
Tag Singulation Fields		The remaining, fields depending on the value of Select Options.
Embedded Command Count	[1 byte]	The number of embedded commands to follow. [Only one allowed]
Embedded Command Length	[1 byte]	Length of embedded commands. Follows standard Length value calculation: number of bytes after OpCode.
Embedded Command OpCode	[1 byte]	The OpCode of the embedded command. Currently supports: • Write Tag EPC (23h) • Write Tag Data (24h) • Lock Tag (25h) • Kill Tag (26h) • Read Tag Data (28h) • Gen2 Tag Specific (2Dh) • BlockWrite (2Dh) • BlockPermaLock (2Eh) • BlockErase (2Eh) When embedding Read Tag Data the complete set of data requested is returned for the first tag that responds with the command response, but up to the first 32 bytes of data requested is returned for every tag that responds and is stored in the tag buffer with the other metadata. See Example with Embedded Read Tag Data below for details.
The fields and values required by the embedded command.		The embedded commands do not support Tag Singulation as it is already performed during the inventory operation. The Options field for the embedded command **must be 0x00**. Note: The **Timeout** field for embedded commands **must be 0x0000.**

Read Tag Multiple Embedded Response Fields

Field	Value	Description
Status	[2 bytes]	Error Code if command failed, otherwise 0x0000 for Success
Select Options	[1 byte]	Options set in the Request Command
Search Flags	[2 bytes]	Search Flags set in the Request Command
Tags Found	[1 byte]	Number of tags found and added to the Tag Buffer matching Select criteria. Follows the standard criteria of adding tags the Tag Buffer. if the Tag Buffer already has tags in it they will not be counted towards Tags Found. However, if they match the Select Criteria they will have the operation performed on them. For this reason it is important to always make sure the Tag Buffer is clear before any Read Tag Multiple execution to insure accurate response information.
Embedded Command Count	[1 byte]	The number of embedded commands to follow. [Currently only supports one]
Embedded Command OpCode	[1 byte]	The OpCode of the embedded command as specified in the request command.
Operations Succeeded	[2 bytes]	Number of Embedded command operations which succeeded. Note: Depending on the Gen2 Session/User Mode used the Operations Succeeded/Failed counts can be misleading since in Session 0, for example, the tag may respond many times during an inventory round and the command may be attempted many times. This would result in counts higher than the actual number of tags the operation succeeded or failed on.
Operations Failed	[2 bytes]	Number of Embedded command operations which failed. Note: As noted above this number can be indicating the command failed multiple times on the same tag. These values should be used in combination with Tag Found and checking the Tag Buffer to insure the operation was completed on the desired tags.
The fields and values returned by the embedded command.		

Example with Embedded Write Tag Data

Here is an example request and response showing the use of Tag Singulation/Select Functionality to inventory tags which meet a specific criteria and then setting the Access password on each using Write Tag Data (24h) as an embedded command:

This command will inventory all tags with an EPC value ending in 0x34, which requires adding the appropriate Tag Singulation Fields to the Basic Tag Inventory syntax then will use

Write Tag Data (24h) to write 0x12345678 into the Reserved Memory Bank starting at Word address 0x00000002 (Access Password)

FF	1E	22	04	00 04	03 E8	00 00 00 00	00 00 00 78	08	34
SOH	Length	OpCode	Options (EPC Mem)	Search Flags	Timeout	Access Password	Select Data Address (bits)	Select Data Length (bits)	Select Data

01	0C	24	00 00	00	00 00 00 02	00	12 34 56 78	AF 29
Embd Cmd Count	Embd Cmd Length	Embd Cmd OpCode	Embd Cmd Timeout (Not Used)	Embd Cmd Options (Must be 0x00)	Write Address (Words)	Write Mem-Bank	Write Data	CRC

The response contains the number of tags found matching the Select criteria specified and the number of embedded command operations which succeeded and failed. Use Get Tag Buffer (29h) to access the tag EPCs and Tag Read Meta Data for the Tags Found. Tags in the buffer may or may not have had successful execution of the embedded command on them:

FF	0A	22	00 00	04	00 04	02	01	24	00 02	00 00	FF 5E
SOH	Length	OpCode	Status	Options	Search Flags	Tag Found	Embd Cmd Count	Embd Cmd OpCode	Operations Succeeded	Operations Failed	CRC

Note

Depending on the Gen2 Session/User Mode used the Operations Succeeded/Failed counts can be misleading since in Session 0, for example, the tag may respond many times during an inventory round and the command may be attempted many times. This would result in counts higher than the actual number of tags the operation succeeded or failed on. The above commands were run in User Mode = Portal.

Note

When embedding write operations, including Lock and Kill, in Read Tag Multiple the Read TX Power will be used for the entire operations: inventory and the write. The power will not switch to the Write TX Power for each Write operation.

Example with Embedded Read Tag Data

When using Read Tag Multiple with an embedded Read Tag Data command an extra field containing the requested Data from the first tag which responded matching the Select criteria is added to the response. Up to 4 bytes of tag data for the other tags responding, if any, is available as Tag Read Meta Data. This command provides the added benefit over simply using just a Read Tag Data with Select of being able to identify other tags matching the criteria which still may need to be written or which you didn't expect to match. An example of sending an embedded Read Tag Data and its response is as follows:

This command will inventory all tags with an EPC value ending in 0x34, which requires adding the appropriate Tag Singulation Fields to the Basic Tag Inventory syntax then will use Read Tag Data (28h) to verify that 0x12345678 was written into the Reserved Memory Bank starting at Word address 0x00000002 (Access Password):

FF	1B	22	04	00 04	03 E8	00 00 00 00	00 00 00 78	08	34
SOH	Length	OpCode	Options (EPC Mem)	Search Flags	Timeout	Access Password	Select Data Address (bits)	Select Data Length (bits)	Select Data

01	09	28	03 E8	00	00	00 00 00 02	02	71 FE
Embd Cmd Count	Embd Cmd Length	Embd Cmd OpCode	Embd Cmd Timeout (Not Used)	Embd Cmd Options (Must be 0x00)	Read Mem-Bank	Read Address (Words)	Read Word Count	CRC

The response contains the number of tags found matching the Select criteria specified and the number of embedded command operations which succeeded and failed. Use Get Tag Buffer (29h) to access the tag EPCs and Tag Read Meta Data for the Tags Found. Tags in the buffer may or may not have had successful execution of the embedded command on them:

FF	0e	22	00 00	04	00 04	03	01	28	00 01	00 00
SOH	Length	OpCode	Status	Options	Search Flags	Tag Found	Embd Cmd Count	Embd Cmd OpCode	Operations Succeeded	Operations Failed

11 22 33 44	DE 02
Data Read	CRC

Example with Embedded Kill Tag

This command will inventory all tags with an EPC value with the first 88bits equal to 0x30940425C4C1967400004E, which requires adding the appropriate Tag Singulation Fields to the Basic Tag Inventory syntax then will use Kill Tag (26h) to kill all matching tags, assuming they have their Kill password set to 0x12345678:

FF	20	22	01	00 04	00 FA	87 65 43 21	58	30 94 04 25 C4 C1 96 74 00 00 4E
SOH	Length	OpCode	Options (EPC Mem)	Search Flags	Timeout	Access Password	Select Data Length (bits)	Select Data

01	08	26	00 00	00	12 34 56 78	00	5F C4
Embd Cmd Count	Embd Cmd Length	Embd Cmd OpCode	Embd Cmd Timeout (Not Used)	Embd Cmd Options (Must be 0x00)	Kill Password	RFU	CRC

Note

If the tags being inventoried and operated on with the embedded command have their Access Password set then the Access Password field must be set accordingly, otherwise it can be left as zero.

The response contains the number of tags found matching the Select criteria specified and the number of embedded command operations which succeeded and failed. Use Get Tag Buffer (29h) to access the tag EPCs for the Tags Found. Tags in the buffer may or may not have had successful execution of the embedded command on them:

FF	0a	22	00 00	01	00 04	01	01	26	00 02	00 00	D9 80
SOH	Length	OpCode	Status	Options	Search Flags	Tag Found	Embd Cmd Count	Embd Cmd OpCode	Operations Succeeded	Operations Failed	CRC

Note

In this example it shows the number of Tags Found lower than the Operations Succeeded. This is because, as with any Read Tag Multiple execution, if the Tag Buffer already has a specific tag in it, it will not be counted towards Tags Found. However, because it matches the Select Criteria it will have the operation performed on it. For this reason it is important to always make sure the Tag Buffer is clear before any Read Tag Multiple execution.

Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h
- FAULT_MSG_INVALID_PARAMETER_VALUE - 105h
- FAULT_NO_PROTOCOL_DEFINED – 401h
- FAULT_AFE_NOT_ON – 405h
- FAULT_NO_TAGS_FOUND – 400h
- FAULT_TAG_ID_BUFFER_FULL – 601h
- FAULT_ANTENNA_NOT_CONNECTED – 503h
- FAULT_TEMPERATURE_EXCEED_LIMITS – 504h
- FAULT_HIGH_RETURN_LOSS – 505h

Write Tag EPC (23h)

The Write Tag EPC command should be used when updating the EPC value of a tag. It is preferred over using Write Tag Data (24h) because Write Tag EPC will automatically lengthen or shorten the EPC ID, by modifying the PC bits, according to the Tag EPC specified. If Write Tag Data is used, the specified data will be modified but the EPC ID length will not be modified.

Note

Write Tag EPC will always use Gen2 Q=0 when Get Protocol Configuration (6Bh) Q Value set to Dynamic Q. For use with large tag populations a Static Q appropriate for the population should be used to avoid collisions.

The **Write Tag EPC** command takes the following fields:

Write Tag EPC Command Fields

Field	Value	Description
Length	[1 byte]	Number of bytes in the command following the OpCode.
OpCode	0x23	Write Tag EPC
Time Out	[2 bytes]	Command timeout in milliseconds.
Select Options	[1 byte]	The Options value of the Tag Singulation Fields
Access Password	[4 bytes]	The Access Password of the tags expected to be inventoried for the embedded command, if they are locked. If the tags are not locked specify 0x00000000. Note: If operating on locked tags, only tags which meet the Select criteria and matching passwords will get a successful execution of the embedded command. Note: If Select Options=0x00 this field should be omitted.
Tag Singulation Fields		The remaining, fields depending on the value of Select Options.
Tag EPC ID	[M bytes]	Up to 496-bit (depending on EPC Length parameter setting in Set Reader Configuration(9Ah)) tag ID to write to the Tag

The reader sends a Fault Code / ACK response back to the host.

An example of **Write Tag EPC** command sequence of events and format is shown next:

1. Starts a timer on the reader.
2. Wakes the tag.
3. Programs the tag with the EPC ID
4. Reads the tag and verifies if the write succeeded.

Note

The verify operation uses the same power level as the write operation. It does not change to the Read power level.

5. Sends back an ACK if OK or a fault code for timeout or other faults.

FF	00	23	00 00	?? ??
SOH	Length	OpCode	Status	CRC

Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h
- FAULT_MSG_INVALID_PARAMETER_VALUE - 105h
- FAULT_NO_PROTOCOL_DEFINED – 401h
- FAULT_AFE_NOT_ON – 405h
- FAULT_NO_TAGS_FOUND – 400h
- FAULT_ANTENNA_NOT_CONNECTED – 503h
- FAULT_TEMPERATURE_EXCEED_LIMITS – 504h
- FAULT_HIGH_RETURN_LOSS – 505h

Write Tag Data (24h)

For the deprecated version of this command see Appendix D: Write Tag Data (24h).

The **Write Tag Data** command writes to the specified memory bank and data address location within that memory bank of a tag. The tag which will be written to can be specified using the Tag Singulation Fields or, if Option=0x00 of the Tag Singulation Fields is specified, it will attempt to write to the first tag it finds. If no tag is in the field, the memory location doesn't exist or is unwriteable, or the Select criteria cannot be satisfied a fault code is returned.

In addition to the Tag Singulation Fields the Write Tag Data command takes several fields which specify the data which will be written to the tag. These fields are:

Write Tag Data Fields

Field	Value	Description
Write Address	4 bytes	The Address field is the offset in the specified Memory Bank, in 16-bit words, where the contents of the Data field is written. It corresponds to the *WordPtr* argument in the Gen2 specification. Note: Addresses are always zero-based. Specifying 0x00 indicates starting at the first address location.
Write MemBank	1 byte	The MemBank field specifies which of the tag's memory banks the data is to be written to. The values correspond to the Memory Bank values as specified in the *Gen2* specification. They are: 0x00 = Reserved 0x01 = EPC 0x02= TID 0x03 = User Memory
Write Data	N bytes	The data to be written to the tag in Memory bank [MemBank] starting at address [Address].
Access Password	4 byes	The Access Password for the tag, if the tag is locked. For an unlocked tag AccessPwd=0x00000000. Note: When Option=0x00 is specified the Access Password is not used.

Examples

The following example will attempt to write to Reserved Memory to set the Kill password=0x11112222. It will write this data to a tag matching the following criteria for a max timeout of 1000 ms.

Memory Bank = User Memory.

Starting Address = bit 32

Select Data = 0x1234

The Reserved Memory bank is not locked so the Access Password is zero

FF	17	24	03 E8	03	00 00 00 00	00	00 00 00 00	00 00 00 20	10	12 34
SOH	Length	OpCode	Time-out (ms)	Option	Write Address	Write MemBank	Access Password	Select Address	Select Data Length	Select Data

11 11 22 22	3E 71
Write Data	CRC

If Option=0x00 or 0x01 is used then the unused Tag Singulation Fields must be removed from the request.

Note

If Option=00, the write tag data is performed without any select criteria resulting the first tag found being written. In that case there is no way to determine what tag gets written to unless there is only one tag in the RF field.

The next example will attempt to write 4 words to User Memory starting at the second word. It will write this data to a tag matching the following criteria for a max timeout of 1000 ms.

EPC ID= 0x0123456789ABCDEF01234567

The User Memory bank is not locked so the Access Password is zero

FF	21	24	03 E8	01	00 00 00 02	03	00 00 00 00	60
SOH	Length	OpCode	Time-out (ms)	Option	Write Address	Write MemBank	Access Password	Select Data Length

01	23	45	67	89	AB	DC	EF	01	23	45	67	11	11	22	22	00	00	00	00	81	E7
Select Data												Write Data								CRC	

Note: Try reading back the value written in this example with the Read Tag Data Examples showing reading from the same tag.

Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h
- FAULT_MSG_INVALID_PARAMETER_VALUE - 105h
- FAULT_NO_PROTOCOL_DEFINED – 401h
- FAULT_AFE_NOT_ON – 405h
- FAULT_NO_TAGS_FOUND – 400h
- FAULT_PROTOCOL_WRITE_FAILED – 406h
- FAULT_PROTOCOL_INVALID_ADDRESS – 409h
- FAULT_GENERAL_TAG_ERROR – 40Ah
- FAULT_GEN2 PROTOCOL_OTHER_ERROR - 420h
- FAULT_GEN2 PROTOCOL_MEMORY_LOCKED - 424h
- FAULT_ANTENNA_NOT_CONNECTED – 503h
- FAULT_TEMPERATURE_EXCEED_LIMITS – 504h
- FAULT_HIGH_RETURN_LOSS – 505h

Lock Tag (25h)

For the deprecated version of this command see Appendix D: Lock Tag (25h).

The **Lock Tag** command locks the specified memory bank of a tag. The tag which will be locked can be specified using the Tag Singulation Fields or, if Option=0x00 of the Tag Singulation Fields is specified, it will attempt to lock the first tag it finds. If no tag is in the field, the memory location doesn't exist or is unlockable, or the Select criteria cannot be satisfied a fault code is returned.

Note

Using Option=0x05 for the Tag Singulation Fields is not allowed for Lock Tag.

In addition to the Tag Singulation Fields the Lock Tag command takes several fields which specify how the tag is to be locked. These fields are:

Lock Tag Fields

Field	Value	Description
AccessPwd	4 byes	The Access Password for the tag.
Mask Bits[1]	2 bytes	The Mask bits specify which fields should be modified according to the Actions bits. When a Mask Bit =0 the corresponding Action bit is not applied and the current lock setting is retained. When a Mask Bit =1 the corresponding Action bit is applied and the new lock setting is implemented.
Action Bits[1]	2 bytes	The Action bits specify whether to assert or deassert a lock behavior for the associated memory location. Action Bits are only applied if the corresponding Mask Bits =1.

1-The Mask and Action bits correspond to the identically named fields described in Section 6.3.2.10.3.5 of the Gen2 specification.

The values of the Mask and Action bits indicate how a tag is to be locked. The 10 Least Significant Bits of each 16-bit argument are used to indicate the lock behavior for each memory bank (Action Bits) and which of those behaviors to apply (Mask Bits). These bits and their corresponding behaviors are:

	First Byte								Second Byte							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Unused						Kill Pwd		Access Pwd		EPC Mem		TID Mem		User Mem	
Mask	X	X	X	X	X	X	Set?	Set?	Set?	Set?	Set?	Set?	Set?	Set?	Set?	Set?
Action	X	X	X	X	X	X	R/W	Perm	R/W	Perm	W	Perm	W	Perm	W	Perm

For each bit in the Mask field where Set?=1 the corresponding Action bit will be applied and the specified lock setting (R/W, W. Permanent) will be asserted (1) or de-asserted (0). Please see the Gen2 specification for more information on Lock Action functionality.

Note

Operations to lock/unlock memory banks can be combined by passing Mask and Action fields with multiple changes, but each bank must exist and be lockable independently. For example, passing 0xFFFF in the **Mask** field and

0x0000 in the **Action** field unlocks all the memory banks as long as all memory banks exist and are lockable.

Examples

The following example shows an attempt to apply a temporary (not permanent) Write lock on the EPC memory (Option=0x01) of a tag whose EPC ID=0x111122223333444455556666 and whose access password=0x11223344:

FF	18	25	03 E8	01	11 22 33 44	00 20	00 20	60h	11 11 22 22 33 33 44 44 55 55 66 66	9E 7A
SOH	Length	OpCode	Timeout (ms)	Option	Access Password	Mask Bits	Action Bits	Select Data Length	Select EPC ID	CRC

Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h
- FAULT_MSG_INVALID_PARAMETER_VALUE - 105h
- FAULT_NO_PROTOCOL_DEFINED – 401h
- FAULT_AFE_NOT_ON – 405h
- FAULT_NO_TAGS_FOUND – 400h
- FAULT_PROTOCOL_INVALID_ADDRESS – 409h
- FAULT_ANTENNA_NOT_CONNECTED – 503h
- FAULT_TEMPERATURE_EXCEED_LIMITS – 504h
- FAULT_HIGH_RETURN_LOSS – 505h

Kill Tag (26h)

For the deprecated version of this command see Appendix D: Kill Tag (26h).

The **Kill Tag** command kills the specified tag. The tag which will be killed can be specified using the Tag Singulation Fields or, if Option=0x00 of the Tag Singulation Fields is specified, it will attempt to kill the first tag it finds. If no tag is in the field, the kill password is zero, or the Select criteria cannot be satisfied a fault code is returned.

In addition to the Tag Singulation Fields the Kill command takes the tag's Kill Password and an extra 1 byte field for future use (RFU).

Examples

The following example shows an attempt to kill a tag whose EPC ID=0x112233445566778899AA and whose Kill password=0x11112222:

FF	13	26	03 E8	01	11 11 22 22	00	50	11 22 33 44 55 66 77 88 99 AA	DD DB
SOH	Length	OpCode	Timeout (ms)	Option	Kill Password	RFU	Select Data Length	Select EPC ID	CRC

Note

If the tag's kill password is set to 0, the protocol does not allow the tag to be killed. A non-zero kill password must be set (using the **Write Tag Data** command) before the kill command succeeds. The RFU field should be set to 0.

Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h
- FAULT_MSG_INVALID_PARAMETER_VALUE - 105h
- FAULT_NO_PROTOCOL_DEFINED – 401h
- FAULT_AFE_NOT_ON – 405h
- FAULT_NO_TAGS_FOUND – 400h
- FAULT_PROTOCOL_INVALID_ADDRESS – 409h
- FAULT_PROTOCOL_INVALID_KILL_PASSWORD – 40Ch
- FAULT_PROTOCOL_KILL_FAILED - 40Eh
- FAULT_GEN2 PROTOCOL_OTHER_ERROR - 420h
- FAULT_ANTENNA_NOT_CONNECTED – 503h
- FAULT_TEMPERATURE_EXCEED_LIMITS – 504h
- FAULT_HIGH_RETURN_LOSS – 505h

Read Tag Data (28h)

For the deprecated version of this command see Appendix D: Read Tag Data (28h).

The **Read Tag Data** command reads the specified memory bank at data address location within that memory bank of a tag. The tag which will be read can be specified using the Tag Singulation Fields or, if Option=0x00 of the Tag Singulation Fields is specified, it will attempt to read from the first tag it finds. If no tag is in the field, the memory location doesn't exist or is read locked, or the Select criteria cannot be satisfied a fault code is returned.

In addition to the Tag Singulation Fields the Read Tag Data command takes several fields which specify the data which will be read from the tag. These fields are:

Read Tag Data Fields

Field	Value	Description
Read MemBank	1 byte	The MemBank field specifies which of the tag's memory banks the data is to be read from. The values correspond to the Memory Bank values as specified in the *Gen2* specification. They are: 0x00 = Reserved 0x01 = EPC 0x02= TID 0x03 = User Memory
Read Address	4 bytes	The Address field is the offset in the specified Memory Bank, in 16-bit words, to start reading from. It corresponds to the *WordPtr* argument in the Gen2 specification. Note: Addresses are always zero-based. Specifying 0x00 indicates starting at the first address location.
WordCount	1 byte	The number of words (16 bit chunks) of data to read from memory bank [MemBank] starting at offset [ReadAddress].
Access Password	4 byes	The Access Password for the tag, if the tag is read locked. For an unlocked tag AccessPwd=0x00000000.

The Basic syntax which returns only the requested Tag Data is defined in Get Tag Data. With additional Option bits set, Read Tag Data can also return Tag Read Meta Data using the syntax in Get Tag Data and Meta Data.

Get Tag Data

Examples

The following example will attempt to read the Kill Password (the first 2 words) from Reserved Memory. It will read this data from a tag matching the following criteria for a max timeout of 1000 ms.

Memory Bank = User Memory.

Starting Address = bit 32

Select Data = 0x1234

The Reserved Memory bank is not locked so the Access Password is zero

FF	14	28	03 E8	03	00	00 00 00 00	02	00 00 00 00
SOH	Length	OpCode	Time-out (ms)	Option	Read MemBank	Read Address	Word-Count	Access Password

00 00 00 20	10	12 34	D1 E7
Select Address	Select Data Length	Select Data	CRC

If Option=0x00 or 0x01 is used then the unused Tag Singulation Fields and the Access Password (for Option=0x00 only) must be removed from the request.

The response to this **Read Data** command example is:

FF	05	28	00 00	03	11 11 22 22	10 BF
SOH	Length	OpCode	Status	Option	Data	CRC

The next example will attempt to read four words from User Memory starting at the third word (0x0002). It will read this data from a tag matching the following criteria for a max

timeout of 1000 ms. Note the Options field, no metadata is requested unlike the previous example.

EPC ID = 0x0123456789ABCDEF01234567.

The User Memory bank is not locked so the Access Password is zero

FF	1A	28	03 E8	01	03	00 00 00 02	04	00 00 00 00
SOH	Length	OpCode	Time-out (ms)	Option	Read MemBank	Read Address	Word-Count	Access Password

60	01 23 45 67 89 AB CD EF 01 23 45 67	7A C1
Select Data Length	Select Data	CRC

The response to this **Read Data** command example is:

FF	09	28	00 00	01	AA BB CC DD	00 00 00 00	E7 54
SOH	Length	OpCode	Status	Option	Data Read		CRC

Note: Try changing the value at this memory location with the Write Tag Data Examples showing writing to the same tag.

Get Tag Data and Meta Data

In addition to getting the tag data returned you can also get Tag Read Meta Data for the found tag. This version of Read Tag Data requires bit 4 of the Option flag to be set and takes an additional Metadata Flags field which defines what metadata will be returned. The following table lists the supported values for these fields.

Read Tag Data Get Data and Metadata Request Fields

Field	Value	Description
Option	Bit 4=0 (0x0X)	No Metadata flags are specified and Meta Data will not be returned. This is the Get Tag Data syntax. The lower bits (X) are specified as defined by Tag Singulation/Select Functionality.
	Bit 4=1 (0x1X)	Indicates that Metadata flags are to follow and the corresponding Metadata shall be returned with the tag data. The lower bits (X) are specified as defined by Tag Singulation/Select Functionality.
Metadata Flags (to specify more than one OR the values together)	0x0000	When no flags are set no meta data will be returned, only the tag data.
	0x0001	When bit 0 is set the Read Count will be returned
	0x0002	When bit 1 is set the LQI/RSSI will be returned
	0x0004	When bit 2 is set the Antenna ID will be returned
	0x0008	When bit 3 is set the Frequency will be returned
	0x0010	When bit 4 is set the Timestamp will be returned
	0x0020	When bit 5 is set the RFU (T*hingMagic Only*) will be returned
	0x0040	When bit 6 is set the Protocol ID will be returned.
	0x0080	When bit 7 is set Tag Data information will be returned. Tag Data is always returned for Read Tag Data, this field cause an extra 2 bytes, always 0x0000, to be returned for Tag Data Length
These fields are followed by the Read Tag Data Fields then the Tag Singulation/Select Functionality (set appropriate bits in the Option field defined above, do not specified an additional Option field), used the same as defined in the Get Tag Data syntax, as necessary.		

A response can contain the following information:

Read Tag Data Get Data and Metadata Response Fields

Field	Length	Value
SOH	1 byte	0xFF
Length	1 byte	Based on data returned
OpCode	1 byte	0x28

Field	Length	Value
Status	2 bytes	0x0000 for success. Otherwise see Appendix C: Error Messages
Options	1 byte	As sent in request
Metadata Flags	2 bytes	Metadata contained in response
Read Count[1]	1 byte	Tag EPC/Antenna Read Count
RSSI[1]	1 byte	Return Signal Strength Indicator
Antenna ID[1]	1 byte	Antenna ID, 4 MSBs for TX and 4 LSBs for RX
Frequency[1]	3 bytes	Frequency in kHz
Timestamp[1]	4 bytes	RTC Timestamp
RFU[1]	2 bytes	Reserved for Future Use - ThingMagic Only
Protocol ID[1]	1 byte	Protocol ID of tag (always 0x05 for M5e/M5e-C)
Tag Data Length[1]	2 bytes	N/A - always zero for Read Tag Data.
Tag Data	N bytes	Number of bytes of Tag Data requested.
CRC	2 bytes	Message CRC

1 - Conditionally returned depending on the Metadata Fields specified in the request.

Examples

The following example will attempt to read the Access Password (the first 2 words) from Reserved Memory. It will read this data from a tag matching the following criteria for a max timeout of 1000 ms.

Memory Bank = EPC.

Starting Address = bit 120 (beginning of the last byte of the EPC value: 16 PC bits + 16 CRC bits + 88 EPC bits)

Select Data = 0x34

The Reserved Memory bank is not locked so the Access Password is zero

FF	15	28	03 E8	14	00 14	00	00 00 00 02	02	00 00 00 00
SOH	Length	OpCode	Time-out (ms)	Option	Meta Data Flags	Read MemBank	Read Address	Word-Count	Access Password

00 00 00 78	08	34	9C 0E
Select Address	Select Data Length	Select Data	CRC

The response to this **Read Data** command example is:

FF	0C	28	00 00	14	00 14	22	00 00 00 15	12 34 56 78	B2 B4
SOH	Length	OpCode	Status	Option	Meta Data Flags	Antenna ID	Timestamp	Tag Read Data	CRC

Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h
- FAULT_MSG_INVALID_PARAMETER_VALUE - 105h
- FAULT_NO_PROTOCOL_DEFINED – 401h
- FAULT_AFE_NOT_ON – 405h
- FAULT_NO_TAGS_FOUND – 400h
- FAULT_PROTOCOL_INVALID_ADDRESS – 409h
- FAULT_GENERAL_TAG_ERROR – 40Ah
- FAULT_GEN2 PROTOCOL_OTHER_ERROR - 420h

- FAULT_GEN2 PROTOCOL_MEMORY_LOCKED - 424h
- FAULT_ANTENNA_NOT_CONNECTED – 503h
- FAULT_TEMPERATURE_EXCEED_LIMITS – 504h
- FAULT_HIGH_RETURN_LOSS – 505h

Get Tag Buffer (29h)

After a **Read Tag Multiple** command is executed, the found tags are stored in an internal Tag Buffer. The **Get Tag Buffer** command can perform several different operations depending on the syntax used. These operations are:

- Get tags remaining in the tag buffer
- Get tag EPCs
- Get tag EPCs and their Tag Read Meta Data.

Get Tags Remaining

To determine the number of tags remaining in the buffer, send the **Get Tag Buffer** command with a data length of zero:

FF	00	29	1D 26
SOH	Length	OpCode	CRC

This command returns the current read index, the location of the next tag to be read, and the current write index, the location where the next tag will be written. These two numbers can be used to get the number of tags left in the tag buffer:

```
Tags Left = WriteIndex - ReadIndex
```

The following response shows there are three tags left in the buffer, and the first one has already been read (the read index parameter starts counting from 0.):

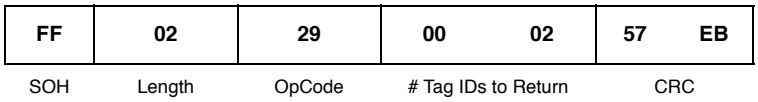
FF	04	29	00 00	00 01	00 04	87 72
SOH	Length	OpCode	Status	ReadIndex	WriteIndex	CRC

Get Tag EPCs

When you want to get the tag EPCs out of the buffer and don't care about the tag read metadata, the options available with this syntax maybe useful. This syntax reads out the requested number of tags from the buffer. A maximum of 13 tags are read at a time, less if Max EPC Length is set to 496 bits. Multiple **Get Tag Buffer** commands may be required to obtain the complete results from a single **Read Tag Multiple** command. If more tags are requested than remain in the buffer an error will be returned.

All tag buffer indexes are encoded as 16-bit unsigned integers. The indexes start counting from 0. Thus, a start index of 0 indicates tag #1, and a start index of 10 would indicate tag #11.

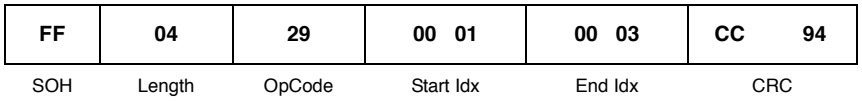
There are two ways to read tag EPCs out of the buffer when you only want EPC values. The first way is to send a **Get Tag Buffer** command with the desired number of tags, *n.* This retrieves the next *n* tags, starting from the current read index. In the previous example, the read index was 1, indicating that one tag was already read. To read the next two tags, set the number of tags parameters to 2. This returns tags number 2 and 3 in the tag buffer:



FF	02	29	00 02	57 EB
SOH	Length	OpCode	# Tag IDs to Return	CRC

When using the **Get Tag Buffer** command in this way, the read index is automatically incremented internally. Thus, if the read index was '1' before getting two tags, then it is incremented to '3' at the end of this command.

Another way to get the second and third tags is to explicitly send the start and end indexes of the tags to read. This is used to retrieve any contiguous block of tag buffer entries at any time.



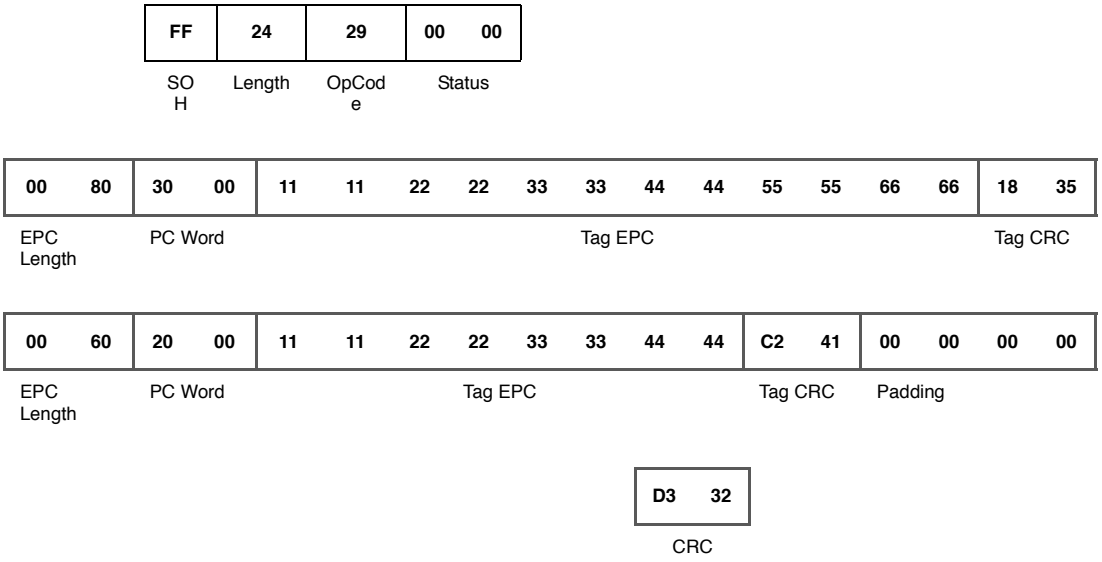
FF	04	29	00 01	00 03	CC 94
SOH	Length	OpCode	Start Idx	End Idx	CRC

Using the start and end index method does not affect the read index that is internally stored in the module. The above request returns a message with two tags. The length of each tag EPC record returned is defined based on the max EPC length configured with Set Reader Configuration(9Ah), regardless of the size of a specific tag's EPC. The sizes and fields of the EPC portion of a tag buffer entry is defined in the Tag Buffer Entry table.

For example:

- A 64-bit tag has the length set to 0x60 (16-bit PC + 64-bit tag EPC + 16-bit tag CRC), but only the first 12 bytes of the tag record is filled in, the trailing bytes, after the CRC are padded with zeros.
- An 96-bit GEN2 tag, the length is 0x80 (16-bit PC (Protocol Control) word + 96-bit tag EPC + 16-bit tag EPC CRC).

In the response below, two tag EPCs are returned. Tag #1 is a 96-bit EPC tag and tag #2 is a 64-bit EPC tag.



Note

Using this syntax the number of tags in the response is not specified. It must be determined by the message length.

Get Tag EPCs and Metadata

Using this syntax for getting information from the tag buffer provides several benefits:

- When at tag EPC is returned there is no padding if the actual tag EPC is shorter than the configured max EPC length. This helps minimize the amount of data returned.
- Any or all fields of the Tag Read Meta Data can be returned.

- In the event of a communication error the last Get EPC and Metadata request can be repeated.

When data is requested using this syntax the response will contain as many tags as can be fit in the response packet. The response will indicate how many tags were returned and should be processed accordingly.

This version of Get Tag Buffer takes two additional fields: the Metadata Flags which defines what metadata will be returned and the Read Options which specifies special read functionality. The following table lists the supported values for these fields.

Get EPC and Metadata Request Fields

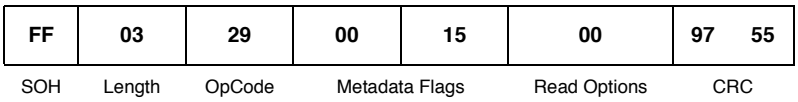
Field	Value	Description
Metadata Flags (to specify more than one OR the values together)	0x0000	When no flags are set no meta data will be returned, only the tag EPC (including PC bits and tag CRC)
	0x0001	When bit 0 is set the Read Count will be returned
	0x0002	When bit 1 is set the LQI/RSSI will be returned
	0x0004	When bit 2 is set the Antenna ID will be returned
	0x0008	When bit 3 is set the Frequency will be returned
	0x0010	When bit 4 is set the Timestamp will be returned
	0x0020	When bit 5 is set the RFU (T*hingMagic Only*) will be returned
	0x0080	When bit 7 is set up to 4 bytes of tag data will be returned. Note: This data is only available if the tag buffer entries are from a Read Tag Multiple (22h) with embedded Read Tag Data (28h).
Read Option	0x00	Read the next set of tags from the Tag Buffer
	0x01	Re-send the last set of tags. Note: When setting this option the tags will be resent starting at the same ReadIndex as the last command. If the metadata requested is different than the last request the number of tags returned might be different. To get those 'missing' tags, additional Get Tag Buffer commands should be sent without the re-send option, otherwise it will reset the ReadIndex again.

Examples

An example command requesting Read Count, AntennaID and Timestamp

```
Metadata Flags = 0x0001 OR 0x0004 OR 0x0010 = 0x0015
```

is as follows:



A response contains the following information:

Get EPC and Metadata Response Fields

Field	Length	Value
SOH	1 byte	0xFF
Length	1 byte	Based on data returned
OpCode	1 byte	0x29
Status	2 bytes	0x0000 for success. Otherwise see Appendix C: Error Messages
Metadata Flags	2 bytes	Metadata contained in response
Read Options	1 byte	As sent in request
Tag Count	1 byte	Number of tags in response
Read Count$_1$	1 byte	Tag EPC/Antenna Read Count
RSSI$_1$	1 byte	Return Signal Strength Indicator
Antenna ID$_1$	1 byte	Antenna ID, 4 MSBs for TX and 4 LSBs for RX
Frequency$_1$	3 bytes	Frequency in kHz
Timestamp$_1$	4 bytes	RTC Timestamp
RFU$_1$	2 bytes	Reserved for Future Use - ThingMagic Only
Protocol ID$_1$	1 byte	Protocol ID of tag read (always 0x05 for M5e/M5e-C)

Field	Length	Value
Tag Data Length[1]	2 byte	Length, in bits, of the tag data read for this tag. This value indicates how many bytes (ceiling[bits/8]), up to 32, will follow. *Example:* if the value is 0x1D (29) then 4 bytes will follow: 29/4 = 3.625 -- ceiling(3.625) = **4**.
Tag Data[1]	N bytes	Number of bytes of tag data as specified in *Tag Data Length*
EPC Length	2 bytes	Number of bits in EPC including PC and CRC bits
PC Word	2 bytes	Tag EPC Protocol Control bits
EPC ID	N bytes	Tag EPC.
Tag CRC	2 bytes	Tag EPC CRC
Repeat fields starting at *Read Count* for remaining tags in message as defined by *Tag Count*		
CRC	2 bytes	Message CRC

1 - Conditionally returned depending on the Metadata Fields specified in the request.

Here is an example response to the example request specified above. The response contains two tags as specified in the Tag Count field each with its EPC info and requested tag read metadata: Read Count, AntennaID and Timestamp:

FF	34	29	00 00	00	15	00	02
SOH	Length	OpCode	Status	Metadata Flags		Read Options	Tag Count

22	11	02 50 CE F6	00 80	31 C1	11 11 22 22 33 33 44 44 55 55 66 66	FB 15
Read Count	Ant ID	Timestamp	EPC Length	PC Word	Tag EPC	Tag CRC

OE	11	04 1D 3D 3C	00 80	30 00	05 00 00 00 00 00 00 00 00 00 23 54	4A C8
Read Count	Ant ID	Timestamp	EPC Length	PC Word	Tag EPC	Tag CRC

1A	B8
CRC	

Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h
- FAULT_MSG_INVALID_PARAMETER_VALUE - 105h
- FAULT_TAG_ID_BUFFER_NOT_ENOUGH_TAGS_AVAILABLE – 600h
- FAULT_TAG_ID_BUFFER_NUM_TAG_TOO_LARGE – 603h

Clear Tag Buffer (2Ah)

The **Clear Tag Buffer** command resets the tag buffer. This clears the buffer of any current tags and reset the read index to 0. A **Read Tag Multiple** command must be issued to load new tags into the buffer.

FF	00	2A	1D	25
SOH	Length	OpCode	CRC	

If **Clear Tag Buffer** or **Get Tag Buffer** is not used after a **Read Tag Multiple** to remove all tags, old tags will be left in the buffer. This means that if tags that have not been removed from the buffer are read again they will not be added to the buffer as second time, only their Read Count will be incremented.

Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h

Gen2 Tag Specific (2Dh)

The **Gen2 Tag Specific** command is a generic command providing an interface for operations proprietary to certain Gen2 tag silicon. The supported commands are grouped by tag silicon type, indicated by the **Chip Type** field. Each Chip Type may have multiple unique commands associated with it, indicated by the **Sub Command** field. Custom commands for the following Chip Types are supported:

- Alien Higgs Silicon (Chip Type=0x01)
- Alien Higgs 3 Silicon (Chip Type=0x05)

- NXP G2X* Silicon (Chip Type=0x02)
- NXP G2i* Silicon (Chip Type=0x07)
- Impinj Monza 4 Silicon (Chip Type=0x08)
- IDS SL900A (Chip Type=0x0A)
- Hitachi Hibiki (Chip Type=0x06) [Deprecated]

Note

Most of the Gen2 Tag Specific commands can be used as an embedded operation in Tag Inventory With Embedded Operations commands. The exceptions are noted in the specific Chip Type command definitions.

Note

The syntax for the Gen2 Tag Specific commands changed in M5e firmware v1.5. The legacy format is still supported but all new development should follow the syntax in this document. For info on the deprecated syntax please contact support@thingmagic.com

Alien Higgs Silicon (Chip Type=0x01)

Tags with Alien Higgs and Higgs-2 (only 96-bit EPC with no user memory versions) Silicon support the following proprietary commands:

Alien Higgs Sub Commands

Sub Command value	Alien Higgs Command
0x01	Partial Load Image
0x03	Full Load Image

Note

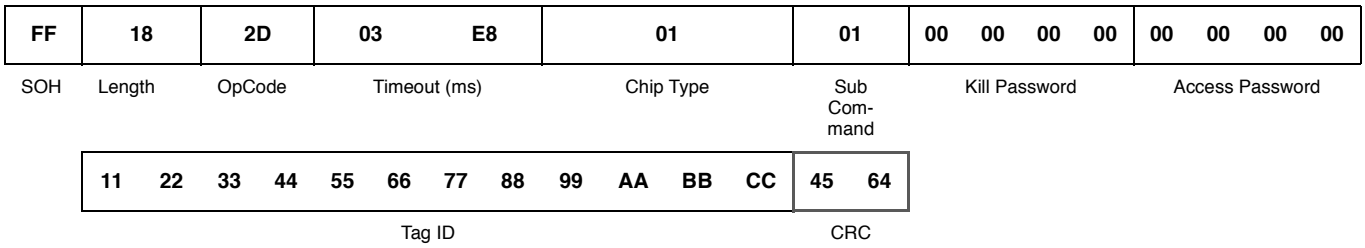
Higgs and Higgs 2 custom commands are not supported as embedded operations in Tag Inventory With Embedded Operations commands. Nor do they support Tag Singulation/Select Functionality.

Partial Load Image

Sub Command=0x01

If the first bit of Sub Command is 1 (Sub Command=0x01), the **Gen2 Tag Specific** command writes an EPC with a length of up to 96-bits, plus the Kill and Access passwords without locking in a single command.

The format for this command is as follows:

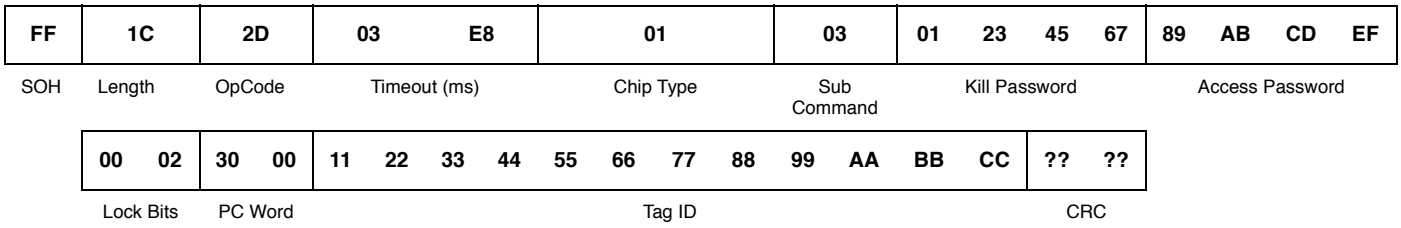


Full Load Image

Sub Command=0x03

If the second bit of Sub Command is 1 (Sub Command=0x03), the **Gen2 Tag Specific** command will also modify the Lock bits (locking the tag according to the Alien Higgs Lock Bits) and the Alien Higgs PC Bits.

The format is extended as follows:



The Lock Bits correspond to the Higgs TID Bank configuration bits with the same name and are shown in the following table:

Alien Higgs Lock Bits

Bit Number	15	14	13	12	11	10	9	8	7	6
Field Name>	X	X	X	X	X	X	X	X	X	X

Default Values	X	X	X	X	X	X	X	X	X	X
Lock Values	X	X	X	X	X	X	X	X	X	X

5	4	3	2	1	0
APW Lock	APW P-lock	KPW Lock	KPW P-lock	EPC Lock	EPC P-lock
0	0	0	0	0	0
1	1	1	1	1	1

Currently, only the first six bits of this field are used. A value of zero in each bit field keeps the corresponding memory segment unlocked. To lock any of the **Kill** and **Access** passwords or the EPC, temporarily or permanently, set the corresponding bit fields in the Lock Bits word to 1. For additional information, refer to the Higgs Tags Application Notes (Currently available at http://www.alientechnology.com/docs/Load_Image_Application_Note_1.pdf - Contact Alien Technology for more details).

The PC Word field, in the table on the previous page, corresponds to the PC Word in the Tag EPC memBank defined in the Gen2 Specification and is shown in the following table:

Alien Higgs PC Bits

Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name	PC Bits					RFU	RFU					NSI	Bits			
Default Values	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0

The PC Word in the **Write Tag Specific** command allows you to specify the NSI Bits and the length of the EPC that you want to write into the tag. The RFU bits are always zero and the PC Bits are determined by the EPC length as specified in the Gen2 Specification and the Higgs Tags Application Notes.

To use the extended format of the **Gen2 Tag Specific** command but keep the NSI bits intact, set option=0x03. This signals the reader to ignore the NSI bits of the PC Word sent to the reader along with the command.

In the example on the previous page, after the execution of the **Gen2 Tag Specific** command, the reader writes the 96-bit EPC 0x112233445566778899AABBCC, the Kill

Password 0x01234567, the Access Password 0x89ABCDEF into the tag without changing the NSI bits of the tag and locks the EPC memBank.

However, to access and write the NSI bits into the tag, set the third bit of the option field to 1 (option=0x07). In the example, if option is set to 0x07, the NSI bits are overwritten by zeros.

The following table summarizes the option field values and the action that is taken:

Bit	Number		
2	**1**	**0**	**Action**
0	0	0	Invalid Value
0	0	1	**Gen2 Tag Specific** without locking and without NSI bits being overwritten
0	1	0	Invalid Value
0	1	1	**Gen2 Tag Specific** with locking and without NSI bits being overwritten
1	0	0	Invalid Value
1	0	1	**Gen2 Tag Specific** without locking and with NSI bits being overwritten by zeros
1	1	0	Invalid Value
1	1	1	**Gen2 Tag Specific** with locking and with NSI bits being overwritten by values that you specify

Alien Higgs 3 Silicon (Chip Type=0x05)

Tags with Alien Higgs 3 Silicon support the following proprietary commands as specified in the *Alien Technology Higgs 3 IC Custom Commands Application Note - Sept. 2008.* (please contact Alien for further details).

Alien Higgs3 Sub Commands

Sub Command value	Alien Higgs3 Command
0x0001	FastLoadImage
0x0003	LoadImage

Sub Command value	Alien Higgs3 Command
0x0009	BlockReadLock

FastLoadImage

Sub Command=0x0001

This command writes all of the following data to the Higgs3 tags in a single command, thereby reducing the tag programming time compared to the use of LoadImage or multiple Write Tag Data (24h) commands.

- Access Passwords
- Kill Password
- EPC Data (96 bits only)

C A U T I O N !

The FastLoadImage command automatically erases the content of all User Memory. If this is undesirable then use the LoadImage or multiple Write Tag Data (24h) commands.

FastLoadImage Command Fields

Field	Value	Description
Length	0x1E	Number of bytes in the command following the OpCode.
OpCode	0x2D	Gen2 Tag Specific Command.
Time Out	[2 bytes]	Command timeout in milliseconds.
Chip Type	0x05	Alien Higgs3.
Tag Singulation Fields I Select Option	[1 byte]	• bit 6 (x1xx xxxx I 0x40) must be set, indicating a 2-byte sub command follows this byte, and before the remaining Tag Singulation Fields
Sub Command	0x0001	FastLoadImage
Tag Singulation Fields I Select [Address, Data Length, Data]	[n bytes]	The remaining fields from the Tag Singulation Fields Note: Optional depending on the value of Select Option.
Current Access Password	[4 bytes]	The Access Password for the tag. If Access Password is currently set and the tag locked, the Access Password must be passed to allow writing.
Kill Password	[4 bytes]	The new Kill Password to be written to Reserved Memory
New Access Password	[4 bytes]	The new Access Password to be written to Reserved Memory
PC Word	[2 bytes]	The value of the PC Word to be written to EPC Memory
EPC Data	[12 bytes]	The 96 bit EPC value to be written to the tag.

An example FastLoadImage single tag command is:.

FF	25	2D	03 E8	05	44	00 01	00 00 00 20	10	11 22
SOH	Length	OpCode	Timeout (ms)	Chip Type	Option	Sub Cmd	Select Data Address	Select Length	Selct Data

11 11 22 22	00 00 00 00	11 22 33 44	XX XX	(12 bytes)	?? ??
Current Access Pwd	Kill Password	New Access Password	PC Word	EPC Data	CRC

The response to the FastLoadImage command is an ACK with the written EPC data. The

format of the response is:

FF	09	2D	00 00	05	44	00 01	XX XX ...	?? ??
SOH	Length	OpCode	Status	Chip Type	Option	Sub Cmd	EPC Data	CRC

FastLoadImage embedded in a Read Tag Multiple takes the following form:

FF	19	22	04	00 04	03 E8	11 22 33 44	00 00 00 78	08	38
SOH	Length	OpCode	Options (EPC Mem)	Search Flags	Timeout	Access Password	Select Data Address (bits)	Select Data Length (bits)	Select Data

01	1F	2D	00 00	05	40	00 01
Embd Cmd Count	Embd Cmd Length	Embd Cmd OpCode	Embd Cmd Timeout (Not Used)	Chip Type	Option	Sub Cmd

11 11 22 22	00 00 00 00	11 22 33 44	XX XX	(12 bytes)	?? ??
Current Access Pwd	Kill Password	New Access Password	PC Word	EPC Data	CRC

LoadImage

Sub Command=0x0003

This command writes all of the following data to the Higgs3 tags in a single command, thereby reducing the tag programming time compared to the use of multiple Write Tag Data (24h) commands.

- Reserved Memory
- EPC Memory
- User Memory

LoadImage Command Fields

Field	Value	Description
Length	0x5E	Number of bytes in the command following the OpCode.
OpCode	0x2D	Gen2 Tag Specific Command.
Time Out	[2 bytes]	Command timeout in milliseconds.
Chip Type	0x05	Alien Higgs3.
Tag Singulation Fields I Select Option	[1 byte]	• bit 6 (x1xx xxxx I 0x40) must be set, indicating a 2-byte sub command follows this byte, and before the remaining Tag Singulation Fields
Sub Command	0x0003	LoadImage
Tag Singulation Fields I Select [Address, Data Length, Data]	[n bytes]	The remaining fields from the Tag Singulation Fields Note: Optional depending on the value of Select Option.
Current Access Password	[4 bytes]	The Access Password for the tag. If Access Password is currently set and the tag locked, the Access Password must be passed to allow writing.
Kill Password	[4 bytes]	The new Kill Password to be written to Reserved Memory
New Access Password	[4 bytes]	The new Access Password to be written to Reserved Memory
PC Word	[2 bytes]	The value of the PC Word to be written to EPC Memory
EPC Data and User Data	[76 bytes]	The 96 bit EPC value to be written to the tag. Note: The specific mapping of these 76 bytes to the tag EPC or User Memory locations is defined in the *Alien Technology Higgs 3 IC Custom Commands Application Note - Sept. 2008.*

An example LoadImage single tag command is:.

FF	XX	2D	03 E8	05	44	00 03	00 00 00 20	10	11 22
SOH	Length	OpCode	Timeout (ms)	Chip Type	Option	Sub Cmd	Select Data Address	Select Length	Selct Data

11 11 22 22	00 00 00 00	11 22 33 44	XX XX	(76 bytes)	?? ??
Current Access Pwd	Kill Password	New Access Password	PC Word	EPC + User Data	CRC

The response to the LoadImage command is an ACK with the written EPC data. The format of the response is:

FF	XX	2D	00 00	05	44	00 03	XX XX ...	?? ??
SOH	Length	OpCode	Status	Chip Type	Option	Sub Cmd	EPC Data	CRC

LoadImage embedded in a Read Tag Multiple takes the following form:

FF	XX	22	04	00 04	03 E8	11 22 33 44	00 00 00 78	08	38
SOH	Length	OpCode	Options (EPC Mem)	Search Flags	Timeout	Access Password	Select Data Address (bits)	Select Data Length (bits)	Select Data

01	5F	2D	00 00	05	40	00 03
Embd Cmd Count	Embd Cmd Length	Embd Cmd OpCode	Embd Cmd Timeout (Not Used)	Chip Type	Option	Sub Cmd

11 11 22 22	00 00 00 00	11 22 33 44	XX XX	(76 bytes)	?? ??
Current Access Pwd	Kill Password	New Access Password	PC Word	EPC + User Data	CRC

BlockReadLock

Sub Command=0x0009

This command allows four-word blocks of User Memory to be read locked. Once read locked the correct Access Password will be required to read the contents of the locked blocks with Read Tag Data (28h).

BlockReadLock Command Fields

Field	Value	Description
Length	0x09	Number of bytes in the command following the OpCode.
OpCode	0x2D	Gen2 Tag Specific Command.
Time Out	[2 bytes]	Command timeout in milliseconds.
Chip Type	0x05	Alien Higgs3.
Tag Singulation Fields I Select Option	[1 byte]	• bit 6 (x1xx xxxx I 0x40) must be set, indicating a 2-byte sub command follows this byte, and before the remaining Tag Singulation Fields
Sub Command	0x0009	BlockReadLock
Tag Singulation Fields I Select [Address, Data Length, Data]	[n bytes]	The remaining fields from the Tag Singulation Fields Note: Optional depending on the value of Select Option.
Current Access Password	[4 bytes]	The Access Password for the tag. If Access Password is currently set and the tag locked, the Access Password must be passed to allow writing.
Command Lock Bits	[1 byte]	Setting a bit value = 1 locks the corresponding User Memory Words. The least significant bit is bit0: bit7=1 Locks User words 0 thru 3 bit6=1 Locks User words 4 thru 7 bit5=1 Locks User words 8 thru 11 bit4=1 Locks User words 12 thru 15 bit3=1 Locks User words 16 thru 19 bit2=1 Locks User words 20 thru 23 bit1=1 Locks User words 24 thru 27 bit0=1 Locks User words 28 thru 31 Note: Depending on the tag memory configuration all 32 Words of User memory may not be available.

An example BlockReadLock single tag command is:.

FF	25	2D	03 E8	05	44	00 09	00 00 00 20	10	11 22
SOH	Length	OpCode	Timeout (ms)	Chip Type	Option	Sub Cmd	Select Data Address	Select Length	Selct Data

11 11 22 22	XX	?? ??
Current Access Pwd	Lock Bits	CRC

The response to the BlockReadLock command is an ACK. The format of the response is:

FF	04	2D	00 00 05	44	00 09	?? ??
SOH	Length	OpCode	Status	Chip Type	Option	Sub Cmd	CRC

BlockReadLock embedded in a Read Tag Multiple takes the following form:

FF	XX	22	04	00 04	03 E8	11 22 33 44	00 00 00 78	08	38
SOH	Length	OpCode	Options (EPC Mem)	Search Flags	Timeout	Access Password	Select Data Address (bits)	Select Data Length (bits)	Select Data

01	0B	2D	00 00	05	40	00 09
Embd Cmd Count	Embd Cmd Length	Embd Cmd OpCode	Embd Cmd Timeout (Not Used)	Chip Type	Option	Sub Cmd

11 11 22 22	XX	?? ??
Current Access Pwd	Lock Bits	CRC

NXP G2X* Silicon (Chip Type=0x02)

Tags with NXP Silicon support the following proprietary commands as specified in the *SL31CS1202 G2XM UCode Functional Specification* (please contact NXP for details):

NXP Sub Commands

Sub Command value	NXP Command
0x0001	Set ReadProtect
0x0002	Reset ReadProtect
0x0003	Change EAS
0x0004	EAS Alarm
0x0005	Calibrate

All NXP-proprietary commands contain the following fields:

NXP Common Fields

Field	Value	Notes
Length	[1 byte]	
OpCode	0x2D	
Timeout	[2 bytes]	Ignored when used as an embedded command for Tag Inventory With Embedded Operations
Chip Type	[1 byte]	• 0x02 - NXP G2X* • 0x07 - NXP G2i*
Tag Singulation Fields l Select Option	[1 byte]	• bit 6 (x1xx xxxx l 0x40) must be set, indicating a 2-byte sub command follows this byte, and before the remaining Tag Singulation Fields
Sub Command	[2 bytes]	NXP Sub Commands
Tag Singulation Fields l Select [Address, Data Length, Data]	[n bytes]	The remaining fields from the Tag Singulation Fields Note: Optional depending on the value of Select Option.
Access Password	[4 bytes]	• Must be non-zero for Set/Reset ReadProtect and Change EAS • EAS Alarm does not take the Access Password field.

In addition to the common fields some commands have additional required fields as indicated below.

Set ReadProtect

The NXP command Set ReadProtect causes the commands Read, Write, Kill, Lock, Access, Set ReadProtect, Change EAS, EAS Alarm and Calibrate to be disabled until a

"Reset ReadProtect" is received. This command uses only the common fields and is invoked with the following command:

FF	10	2D	03 E8	02	44	00 01	00 00 00 78	08	34	11 22 33 44	?? ??
SOH	Length	OpCode	Timeout (ms)	Chip Type	Option	Sub Cmd	Select Data Address	Select Length	Selct Data	Access Password	CRC

The response to the Set ReadProtect command is a simple ACK with standard status:

FF	04	2D	00 00	02	44	00 01	?? ??
SOH	Length	OpCode	Status	Chip Type	Option	Sub Cmd	CRC

Set ReadProtect embedded in a Read Tag Multiple take the following form:

FF	1C	22	04	00 04	03 E8	11 22 33 44	00 00 00 78	08	34
SOH	Length	OpCode	Options (EPC Mem)	Search Flags	Timeout	Access Password	Select Data Address (bits)	Select Data Length (bits)	Select Data

01	0A	2D	00 00	02	40	00 01	11 22 33 44 ?? ??
Embd Cmd Count	Embd Cmd Length	Embd Cmd OpCode	Embd Cmd Timeout (Not Used)	Chip Type	Option	Sub Cmd	Access Password) CRC

Reset ReadProtect

The NXP command Reset ReadProtect restores normal operation to a tag which is in ReadProtect mode. This command uses only the common fields and is invoked with the following command:

Note

Only the G2i command can be performed as an embedded operation in a Read Tag Multiple. The G2x version cannot.

FF	10	2D	03 E8	02	44	00 02	00 00 00 78	08	34	11 22 33 44	?? ??
SOH	Length	OpCode	Timeout (ms)	Chip Type	Option	Sub Cmd	Select Data Address	Select Length	Selct Data	Access Password	CRC

The response to the Reset ReadProtect command is a simple ACK with standard status:

FF	04	2D	00 00 02	44	00 02	?? ??
SOH	Length	OpCode	Status	Chip Type	Option	Sub Cmd	CRC

Reset ReadProtect embedded in a Read Tag Multiple take the following form:

FF	1C	22	04	00 04	03 E8	11 22 33 44	00 00 00 78	08	34
SOH	Length	OpCode	Options (EPC Mem)	Search Flags	Timeout	Access Password	Select Data Address (bits)	Select Data Length (bits)	Select Data

01	0A	2D	00 00	02	40	00 02	11 22 33 44 ?? ??
Embd Cmd Count	Embd Cmd Length	Embd Cmd OpCode	Embd Cmd Timeout (Not Used)	Chip Type	Option	Sub Cmd	Access Password) CRC

Change EAS

The NXP command Change EAS sets or resets the EAS system bit. When set, the tag will return an alarm code if an “EAS Alarm” command is received. This command takes an additional field:

- **Change EAS** (1 byte: 0x01=Set EAS; 0x02=Reset EAS)

It is invoked with the following command:

FF	11	2D	03 E8	02	44	00 03	00 00 00 78	08	34	11 22 33 44	01	?? ??
SOH	Length	OpCode	Timeout (ms)	Chip Type	Option	Sub Cmd	Select Data Address	Select Length	Selct Data	Access Password	EAS	CRC

The response to the ChangeEAS command is a simple ACK with standard status:

FF	04	2D	00 00	02	44	00 03	?? ??
SOH	Length	OpCode	Status	Chip Type	Option	Sub Cmd	CRC

ChangeEAS embedded in a Read Tag Multiple take the following form:

FF	1C	22	04	00 04	03 E8	11 22 33 44	00 00 00 78	08	34
SOH	Length	OpCode	Options (EPC Mem)	Search Flags	Timeout	Access Password	Select Data Address (bits)	Select Data Length (bits)	Select Data

01	0B	2D	00 00	02	40	00 01	11 22 33 44	01	?? ??
Embd Cmd Count	Embd Cmd Length	Embd Cmd OpCode	Embd Cmd Timeout (Not Used)	Chip Type	Option	Sub Cmd	Access Password)	EAS	CRC

EAS Alarm

The NXP command EAS Alarm causes the tag to return coded information, but only when the tag has its EAS system bit set. This command requires several fields in addition to the common fields

- **Divide Ratio as Per Gen2 (DR)** (1 byte: Current fixed to 0x01, M5e and M5e-C only support DR=64/3)
- **Miller Cycles (M)** (1 byte: 0x01, M=2; 0x02, M=4;or 0x03, M=8)

Note: This value of M must be the same as the value set in Set Protocol Configuration (9Bh).

- **TrExt as Per Gen2** (1 byte: Current fixed to 0x01, M5e and M5e-C only support extended Pilot Tone)

It is invoked with the following command, **it does not support tag singulation**:

FF	09	2D	03 E8	02	4o	00 04	01	02	01	?? ??
SOH	Length	OpCode	Timeout (ms)	Chip Type	Option	Sub Cmd	DR	M	TrExt	CRC

The response to the EAS Alarm command contains 8 bytes of EAS Alarm Data, only if Status=0x0000. The format of the response is:

FF	0A	2D	00 00	02	44	00 04	[8 bytes]	?? ??
SOH	Length	OpCode	Status	Chip Type	Option	Sub Cmd	[EAS Alarm Data]	CRC

Note

EASAlarm cannot be embedded in Read Tag Multiple.

Calibrate

The NXP command Calibrate causes the tag to return a random data pattern that is useful in some frequency spectrum measurements.

Calibrate can only be sent when the tag is in the Secured state, when the access password is non-zero. This command uses only the common fields and is invoked with the following command.

FF	10	2D	03 E8	02	44	00 05	00 00 00 78	08	34	11 22 33 44	?? ??
SOH	Length	OpCode	Timeout (ms)	Chip Type	Option	Sub Cmd	Select Data Address	Select Length	Selct Data	Access Password	CRC

The response to the Calibrate command contains 64 bytes of Calibrate Data, only if Status=0x0000. The format of the response is:

FF	09	2D	00 00	02	44	00 05	[64 bytes]	?? ??
SOH	Length	OpCode	Status	Chip Type	Option	Sub Cmd	[Calibrate Data]	CRC

Calibrate embedded in a Read Tag Multiple take the following form:

FF	19	22	04	00 04	03 E8	11 22 33 44	00 00 00 78	08	34
SOH	Length	OpCode	Options (EPC Mem)	Search Flags	Timeout	Access Password	Select Data Address (bits)	Select Data Length (bits)	Select Data

01	0A	2D	00 00	02	40	00 05	11 22 33 44	?? ??
Embd Cmd Count	Embd Cmd Length	Embd Cmd OpCode	Embd Cmd Timeout (Not Used)	Chip Type	Option	Sub Cmd	Access Password)	CRC

Note

If the tag is a G2XL without User Memory the *Calibrate Data* response field will contain 64 bytes of ‘00’.

NXP G2i* Silicon (Chip Type=0x07)

Tags using the NXP G2i Silicon (SL3S1203_1213), please contact NXP for further details, support all the commands defined for NXP G2X* Silicon (Chip Type=0x02), with the caveats as noted in the command descriptions. In addition the following custom commands is G2i* only.

ChangeConfig

The ChangeConfig command is used to toggle the bits of the G2i* tag's ConfigWord (see ConfigWord to Command Bit Address Map table). Specify a '1' in the desired bit location of the Config Data word of the command in order to toggle that bit.

Note

Different version of the G2i* tags support different features. See tag data sheet for specific bits supported.

The ChangeConfig command can ONLY be sent in the Secured state, i.e. requires a non-zero password. Caution should be used when embedding ChangeConfig in a Tag Inventory With Embedded Operations command. Since this command toggles the specified bits, if the tag responds twice (or an even number of times) during an inventory round the end result will be no change.

ConfigWord to Command Bit Address Map

Config Data Bit Address (LSB=bit 0)	G2i* ConfigWord Bit Address (Hex)	Toggle Bit Description
15	0x200	Tamper Alarm Flag (read only)
14	0x201	External Supply Flag Digital Input (read only)
13	0x202	RFU
12	0x203	RFU
11	0x204	Invert Digital Output (reset at power up)
10	0x205	Transparent Mode On/Off (reset at power up)
9	0x206	Transparent Mode data/raw (reset at power up)
8	0x207	Conditional Read Range Reduction (permanently stored in tag memory)
7	0x208	Conditional Range Redcution Open/Short (permanently stored in tag memory)
6	0x209	Maximum Backscatter Strength (permanently stored in tag memory)
5	0x20A	Digital Output (permanently stored in tag memory)
4	0x20B	Read Range Reduction On/Off (permanently stored in tag memory)
3	0x20C	Read Protect User Memory (permanently stored in tag memory)

Config Data Bit Address (LSB=bit 0)	G2i* ConfigWord Bit Address (Hex)	Toggle Bit Description
2	0x20D	Read Protect EPC Bank (permanently stored in tag memory)
1	0x20E	Read Protect TID (permanently stored in tag memory)
0	0x20F	PSF Alarm Flag (permanently stored in tag memory)

NXP G2i* Command Fields

Field	Value	Notes
Length	[1 byte]	
OpCode	0x2D	
Timeout	[2 bytes]	Ignored when used as an embedded command for Tag Inventory With Embedded Operations
Chip Type	0x07	0x07 - NXP G2i*
Tag Singulation Fields l Select Option	[1 byte]	• bit 6 (x1xx xxxx l 0x40) must be set, indicating a 2-byte sub command follows this byte, and before the remaining Tag Singulation Fields
Sub Command	[2 bytes]	• 0x0007 - ChangeConfig
Access Password	[4 bytes]	Must be non-zero for ChangeConfig. If zero the ConfigWord will not be changed.
Tag Singulation Fields l Select [Address, Data Length, Data]	[n bytes]	The remaining fields from the Tag Singulation Fields Note: Optional depending on the value of Select Option.
RFU	0x00	RFU
Config Data	[2 bytes]	Toggle memory bits as defined by ConfigWord to Command Bit Address Map • '1' - Toggle memory bit • '0' - Leave memory alone

An example ChangeConfig single tag command is:.

FF	10	2D	03 E8	07	44	00 07	11 22 33 44	00 00 00 78	08	34
SOH	Length	OpCode	Timeout (ms)	Chip Type	Option	Sub Cmd	Access Password	Select Data Address	Select Length	Selct Data

00	00 00	?? ??
RFU	Config Data	CRC

The response to the ChangeConfig command contains the new value of the tag's ConfigWord after the Config Data toggle bits have been appled. The format of the

response is:

FF	09	2D	00 00	07	44	00 07	00 00	?? ??
SOH	Length	OpCode	Status	Chip Type	Option	Sub Cmd	New Config-Word Value	CRC

ChangeConfig embedded in a Read Tag Multiple takes the following form:

FF	19	22	04	00 04	03 E8	11 22 33 44	00 00 00 78	08	34
SOH	Length	OpCode	Options (EPC Mem)	Search Flags	Timeout	Access Password	Select Data Address (bits)	Select Data Length (bits)	Select Data

01	09	2D	00 00	07	40	00 07	00	00 00	?? ??
Embd Cmd Count	Embd Cmd Length	Embd Cmd OpCode	Embd Cmd Timeout (Not Used)	Chip Type	Option	Sub Cmd	RFU	Config Data	CRC

Impinj Monza 4 Silicon (Chip Type=0x08)

Tags with the Impinj Monza 4 chip support custom functionality providing protection of data through public and private data profiles and the use of short range reading options. These are controlled through custom commands as defined by the Impinj Monza 4 datasheet (IPJ_Monza4Datasheet_20101101.pdf (please contact Impinj for more details.

QT Read/Write

The QT command controls the switching of Monza 4QT between the Private and Public profiles. The tag MUST be in the Secured state to succeed.

Impinj Monza 4QT Command Fields

Field	Value	Notes
Length	[1 byte]	
OpCode	0x2D	
Timeout	[2 bytes]	Ignored when used as an embedded command for Tag Inventory With Embedded Operations
Chip Type	0x08	0x08 - Impinj Monza 4QT
Tag Singulation Fields l Select Option	[1 byte]	bit 6 (x1xx xxxx l 0x40) must be set, indicating a 2-byte sub command follows this byte, and before the remaining Tag Singulation Fields
Sub Command	0x0000	• 0x0000 - QT Read/Write
Access Password	[4 bytes]	• Must be non-zero for QT Read/Write
Tag Singulation Fields l Select [Address, Data Length, Data]	[n bytes]	The remaining fields from the Tag Singulation Fields Note: Optional depending on the value of Select Option.
Control Byte	0xX0	Only the 4 most significant bits are used, as follows: **Bit 7 (MSB) - Read/Write bit** • 0 = read the QT control bits in cache • 1 = write the QT control bits **Bit 6 - Persistence bit** • 0 = write the QT Control to volatile memory • 1 = write the QT Control to nonvolatile memory **Bits 5 and 4 - RFU, currently ignored by tag**
Payload	[2 bytes]	Controls the QT functionality. These bits are ignored when the Read/Write bit =0. **Bit 15 (MSB) - QT_SR** • 0 = Tag does not reduce range • 1 = Tag reduces range if in or about to be in OPEN or SECURED state. **Bit 14 - QT_MEM** • 0 = Tag uses Private Memory Map • 1 = Tag uses Public Memory Map **Bits 13:0 - RFU**

An example QT Read/Write single tag command is:.

FF	09	2D	03 E8	08	40	00 00	00	00 00	?? ??
SOH	Length	OpCode	Timeout (ms)	Chip Type	Option	Sub Cmd	Control Byte	Payload	CRC

The response to the QT Read/Write command contains the payload read or written as specified by the control byte's Read/Write bit. The format of the response is:

FF	06	2D	00 00	08	40	00 00	00 00	?? ??
SOH	Length	OpCode	Status	Chip Type	Option	Sub Cmd	Payload	CRC

QT Read/Write embedded in a Read Tag Multiple takes the following form:

FF	1A	22	04	00 04	03 E8	11 22 33 44	00 00 00 78	08	34
SOH	Length	OpCode	Options (EPC Mem)	Search Flags	Timeout	Access Password	Select Data Address (bits)	Select Data Length (bits)	Select Data

01	08	2D	00 00	08	40	00 00	00	00 00	?? ??
Embd Cmd Count	Embd Cmd Length	Embd Cmd OpCode	Embd Cmd Timeout (Not Used)	Chip Type	Option	Sub Cmd	Control Byte	Payload	CRC

IDS SL900A (Chip Type=0x0A)

Please see the *MercuryAPI Programmers Guide | Level 2 | Advanced Tag Operations* section for the programming interface supporting IDS SL900A "Cool Log" functionality.

For the serial protocol specification for these tags please contact ThingMagic Support.

Hitachi Hibiki (Chip Type=0x06) [Deprecated]

Tags with Hitachi Hibiki Silicon support the following proprietary commands as specified in the *Secure RFID Protocol Specification v.1.34 June 29, 2007.* (please contact Hitachi for details).

Please contact ThingMagic Support for assistance using Hibiki operations.

Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h
- FAULT_MSG_INVALID_PARAMETER_VALUE - 105h
- FAULT_NO_PROTOCOL_DEFINED – 401h
- FAULT_AFE_NOT_ON – 405h
- FAULT_NO_TAGS_FOUND – 400h
- FAULT_PROTOCOL_INVALID_ADDRESS – 409h
- FAULT_GENERAL_TAG_ERROR – 40Ah
- FAULT_GEN2 PROTOCOL_OTHER_ERROR - 420h
- FAULT_ANTENNA_NOT_CONNECTED – 503h
- FAULT_TEMPERATURE_EXCEED_LIMITS – 504h
- FAULT_HIGH_RETURN_LOSS – 505h

BlockWrite (2Dh)

BlockWrite is a Gen2 2.0 Optional command. On tags which support this command, it provides faster writing of data to a tag by writing more than one word at a time.

The BlockWrite command takes the following fields:

BlockWrite Command Fields

Field	Value	Notes
Length	[1 byte]	Number of bytes in the command following the OpCode.
OpCode	0x2D	BlockWrite
Timeout	[2 bytes]	Ignored when used as an embedded command for Tag Inventory With Embedded Operations
Chip Type	0x00	Standard Gen2 2.0 tags
Tag Singulation Fields l Select Option	[1 byte]	bit 6 (x1xx xxxx l 0x40) must be set, indicating a 2-byte sub command follows this byte, and before the remaining Tag Singulation Fields
Sub Command	0x00C7	• 0x00C7 - BlockWrite
Access Password	[4 bytes]	The tag's Access Password for writing to write locked memory
Tag Singulation Fields l Select [Address, Data Length, Data]	[n bytes]	The remaining fields from the Tag Singulation Fields Note: Optional depending on the value of Select Option.
WriteFlags	0x00	Reserved for future use.
MemoryBank	[1 byte]	Memory bank who's data will be erased. Standard Gen2 field values: • 0x00 = Reserved • 0x01 = EPC • 0x02 = TID • 0x03 = User
WordPointer	[4 bytes]	Starting, zero-based, 16-bit word address of the MemoryBank to be written.
WordCount	[1 bytes]	Number of 16-bit words to be written.
Data	[2*WordCount bytes]	Data to be written

An example BlockWrite single tag command is:.

FF	09	2D	03 E8	00	44	00 C7	11 22 33 44	00 00 00 78	08	34
SOH	Length	OpCode	Timeout (ms)	Chip Type	Option	Sub Cmd	Access Password	Select Data Address (bits)	Select Data Length (bits)	Select Data

00	00	00	01	00 00	?? ??
Write Flag	Mem Bank	Word Pointer	Word Count	Data	CRC

The response to the BlockWrite command is an ACK. The format of the response is:

FF	04	2D	00 00	00	44	00 C7	?? ??
SOH	Length	OpCode	Status	Chip Type	Option	Sub Cmd	CRC

BlockWrite embedded in a Read Tag Multiple takes the following form:

FF	1A	22	04	00 04	03 E8	11 22 33 44	00 00 00 78	08	34
SOH	Length	OpCode	Options (EPC Mem)	Search Flags	Timeout	Access Password	Select Data Address (bits)	Select Data Length (bits)	Select Data

01	10	2D	00 00	00	44	00 C7	00	00	00	01	00 00	?? ??
Embd Cmd Count	Embd Cmd Length	Embd Cmd OpCode	Embd Cmd Timeout (Not Used)	Chip Type	Option	Sub Cmd	Write Flag	Mem Bank	Word Pointer	Word Count	Data	CRC

BlockPermaLock (2Eh)

BlockPermaLock is a Gen2 2.0 Optional command. On tags which support this command, it allows User Memory to be selectively, permanently write-locked in individual sub-portions. Compare BlockPermaLock with standard Lock Tag (25h) which only allows locking entire memory banks and allows for no permanent locking.. The block-size is tag-specific. For example Alien Higgs3 tags support 4 word blocks.

The BlockPermaLock command takes the following fields:

BlockPermaLock Command Fields

Field	Value	Description
Length	[1 byte]	Number of bytes in the command following the OpCode.
OpCode	0x2E	Gen2 Optional Command.
Time Out	[2 bytes]	Command timeout in milliseconds.
Chip Type	0x00	Gen2 2.0 Tags with BlockPermalock support
Tag Singulation Fields I Select Option	[1 byte]	bit 6 (x1xx xxxx I 0x40) must be set, indicating a 1-byte sub command follows this byte, and before the remaining Tag Singulation Fields
Sub Command	0x01	BlockPermaLock
Access Password	[4 bytes]	The Access Password for the tag. Note: Do not include password if Select Option = 0xX0.
Tag Singulation Fields I Select [Address, Data Length, Data]	[n bytes]	The remaining fields from the Tag Singulation Fields Note: Optional depending on the value of Select Option.
RFU	0x00	Reserved for future use.
Read/Lock	[1 byte]	Only the Least Significant Bit is used: • 0x00 = Return the PermaLock status of the blocks specified. • 0x01 = PermaLock the blocks specified
MemoryBank	0x03	Memory bank who's blocks will be locked. Currently only User Memory is supported.
BlockPointer	[4 bytes]	The starting address for Mask, in units of 16 blocks. For example, BlockPtr = 0x00 indicates block 0, BlockPtr = 0x01 indicates block 16, BlockPtr = 0x02 indicates block 32. Depending on the tag memory configuration all 32 Words of User memory may not be available. Note: For Higgs3 tags this value must always be 0x00
BlockRange	[1 byte]	The range of Mask, starting at BlockPointer and ending (16×BlockRange)–1 blocks later. Note: For Higgs3 tags this value must always be 0x01
Mask	[2*Block-Range bytes]	Memory blocks to PermaLock. Mask depends on the Read/Lock bit as follows: • Read/Lock = 0: Do not include the Mask field. • Read/Lock = 1: Mask is 16×BlockRange bits. The Mask bits shall be ordered from lower-order block to higher (i.e. if BlockPtr = 00h then the most-significant Mask bit refers to block 0). – Mask bit = 0: No change to block. – Mask bit = 1: Permalock the corresponding memory block.

An example BlockPermaLock single tag command, locking Block 5, is:.

FF	1A	2E	03 E8	00	42	01	11 11 22 22	00 00 00 00	10	E2 00
SOH	Length	OpCode	Timeout (ms)	Chip Type	Option	Sub Cmd	Access Passwords	Select Data Address	Select Length	Selct Data

00	01	03	00 00 00 00	01	04 00	?? ??
RFU	Read/ Lock	Mem Bank	Block Ptr	Block Range	Mask	CRC

The response to the BlockReadLock command contains the current (for just a read operation) or updated (for a lock operation) status of the memory blocks requested. The format of the response is:

FF	XX	2E	00 00	42	XX	XX XX ...	?? ??
SOH	Length	OpCode	Status	Option	Word Count	Tag's Lock Response Mask (2*BlockRange bytes)	CRC

The same BlockPermaLock embedded in a Read Tag Multiple takes the following form:

FF	21	22	02	00 04	03 E8	11 22 33 44	00 00 00 00	08	E2
SOH	Length	OpCode	Options (TID Mem)	Search Flags	Timeout	Access Password	Select Data Address (bits)	Select Data Length (bits)	Select Data

01	0F	2E	00 00	00	40	01
Embd Cmd Count	Embd Cmd Length	Embd Cmd OpCode	Embd Cmd Timeout (Not Used)	Chip Type	Option	Sub Cmd

00	01	03	00 00 00 00	01	04 00	?? ??
RFU	Read/ Lock	Mem Bank	Block Ptr	Block Range	Mask	CRC

BlockErase (2Eh)

BlockErase is a Gen2 2.0 Optional command. On tags which support this command, it allows multiple words in any memory bank to be erased in a single operation.

The BlockErase command takes the following fields:

BlockWrite Command Fields

Field	Value	Description
Length	[1 byte]	Number of bytes in the command following the OpCode.
OpCode	0x2E	Gen2 Optional Command.
Time Out	[2 bytes]	Command timeout in milliseconds.
Chip Type	0x00	Gen2 2.0 Tags with BlockPermalock support
Tag Singulation Fields l Select Option	[1 byte]	bit 6 (x1xx xxxx l 0x40) must be set, indicating a 1-byte sub command follows this byte, and before the remaining Tag Singulation Fields
Sub Command	0x00	BlockErase
Access Password	[4 bytes]	The Access Password for the tag. Note: Do not include password if Select Option = 0xX0.
Tag Singulation Fields l Select [Address, Data Length, Data]	[n bytes]	The remaining fields from the Tag Singulation Fields Note: Optional depending on the value of Select Option.
WordPointer	[4 bytes]	Starting, zero-based, 16-bit word address of the MemoryBank to be erased.
MemoryBank	[1 byte]	Memory bank who's data will be erased. Standard Gen2 field values: • 0x00 = Reserved • 0x01 = EPC • 0x02 = TID • 0x03 = User
WordCount	[1 bytes]	Number of 16-bit words to be erased.

An example BlockErase single tag command is:.

FF	1A	2E	03 E8	00	42	00	11 11 22 22	00 00 00 00	10	E2 00
SOH	Length	OpCode	Timeout (ms)	Chip Type	Option	Sub Cmd	Access Passwords	Select Data Address	Select Length	Selct Data

00 00 00 02	01	01	?? ??
Address	Mem Bank	Word Count	CRC

The response to the BlockErase command is an ACK. The format of the response is:

FF	XX	2E	00 00	?? ??
SOH	Length	OpCode	Status	CRC

The same BlockErases embedded in a Read Tag Multiple takes the following form:

FF	21	22	02	00 04	03 E8	11 22 33 44	00 00 00 00	08	E2
SOH	Length	OpCode	Options (TID Mem)	Search Flags	Timeout	Access Password	Select Data Address (bits)	Select Data Length (bits)	Select Data

01	0F	2E	00 00	00	40	00
Embd Cmd Count	Embd Cmd Length	Embd Cmd OpCode	Embd Cmd Timeout (Not Used)	Chip Type	Option	Sub Cmd

00 00 00 02	01	01	?? ??
Address	Mem Bank	Word Count	CRC

Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h
- FAULT_MSG_INVALID_PARAMETER_VALUE - 105h
- FAULT_NO_PROTOCOL_DEFINED – 401h
- FAULT_AFE_NOT_ON – 405h
- FAULT_NO_TAGS_FOUND – 400h
- FAULT_PROTOCOL_INVALID_ADDRESS – 409h
- FAULT_GENERAL_TAG_ERROR – 40Ah
- FAULT_GEN2 PROTOCOL_OTHER_ERROR - 420h
- FAULT_ANTENNA_NOT_CONNECTED – 503h
- FAULT_TEMPERATURE_EXCEED_LIMITS – 504h
- FAULT_HIGH_RETURN_LOSS – 505h

Set Application Commands

The Set commands are used to set configurable values in the firmware. Since the values are not stored in flash, these values are reset to the default values whenever the application firmware is restarted. The application responds with a fault code / ACK to all commands.

The following table lists the Application Set commands:

Set Application Commands

OpCode	Command Name	Bootloader	App Firmware
0x91	Set Antenna Port (91h)	N	Y
0x92	Set Read TX Power (92h)	N	Y
0x93	Set Current Tag Protocol (93h)	N	Y
0x94	Set Write TX Power (94h)	N	Y
0x95	Set Frequency Hop Table (95h)	N	Y
0x96	Set User GPIO Outputs (96h)	N	Y
0x97	Set Current Region (97h)	N	Y
0x98	Set Power Mode (98h)	N	Y
0x99	Set User Mode (99h)	N	Y
0x9A	Set Reader Configuration(9Ah)	N	Y
0x9B	Set Protocol Configuration (9Bh)	N	Y

Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h
- FAULT_MSG_INVALID_PARAMETER_VALUE - 105h

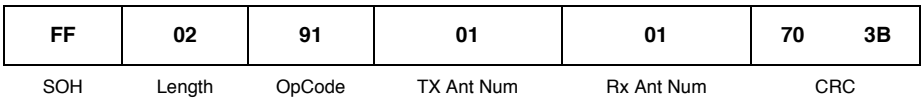
Set Antenna Port (91h)

The Set Antenna Port command supports multiple options (Option=0x01 is not supported) each setting the different aspects of the modules antenna configuration. Please read all the details of various options for setting antenna configuration parameters before using the command.

Set Single Tag Operations Antennas

The **Set Antenna Port** command, using the syntax without an Option field and when Option=0x00, sets the logical transmit (TX) and receive (RX) antennas to be used for Single Tag Operations (Read Tag Single (21h), Write Tag Data (24h), Lock Tag (25h), etc.).

The old syntax, with no Option parameter simply takes a 1 byte TX and 1 byte RX logical antenna number. Valid TX and RX values range from 1 to 8 (assuming two four port multiplexers are used).



All other settings will result in an error.

Using the new syntax with Option = 0x00 duplicates the functionality of the old syntax:

Set Single Tag Ops Antennas Command Fields (Option = 0x00)

Field	Value	Description
Length	[1 byte]	Number of bytes after OpCode
OpCode	0x91	Set Antenna Port
Option	0x00	Set TX / RX Antennas for Single Tag Operations
TX Port	[1 byte]	Specifies which logical antenna to be used for Transmitting
RX Port	[1 byte]	Specifies which logical antenna to be used for Receiving

Note

If a single tag operations antenna is not specified the default is a single bistatic antenna on TX=1/RX=2.

Set Multi-Antenna Search Configuration

Option = 0x02 allows the order antennas will be used during a multi-antenna search with Read Tag Multiple (22h) to be specified. The multi-antenna search allows mixing of monostatic and bistatic configurations and ports can be duplicated in the search list if it is desirable to return to some antennas more often than others. The number of antennas available for use and the specific physical to logical port mapping is based on the *GPIO as Antenna Switch* settings specified in Set Reader Configuration(9Ah).

Set Logical Antennas Command Fields (Option = 0x02)

Field	Value	Description
Length	[1 byte]	Number of bytes after OpCode
OpCode	0x91	Set Antenna Port
Option	0x02	The logical antenna search configuration to follow.
TX/RX Logical Antenna	[2 bytes]	One or multiple 2 byte pairs indicating the logical TX and RX antennas to use. The order specified indicates the order antennas will be used in a multi-antenna search. Note: Monostatic and bistatic logical antennas can be mixed in the same search list and entries can be repeated.
Note: Repeat the TX/RX Logical Antenna field for each logical antenna being configured for the search list in the desired search order.		

Note

If a search configuration is not specified the default is a single bistatic antenna on TX=1/RX=2.

Set Antenna's Power and Settling Time

Option =0x03 and 0x04 allows the power and settling time (only option=0x04) for each logical antenna to be set. The order the antennas settings are defined does not affect search order.

Note

Settling time is the time between once the control lines switch to the next antenna setting and RF turns on for operations on that port. This allows time for external multiplexer's to fully switch to the new port before a signal is sent, if necessary. *Default value is 0*.

C A U T I O N !

When the TX power for a logical antenna is defined using this command the power setting specified for each port specified will be used for *all* operations, overriding the default, module wide, value set in Set Read TX Power (92h) and Set Write TX Power (94h). If a logical antenna's power is not set with this command then it will use the module-wide value set in those commands.

Set Antenna's Power and Settling Command Fields (Option = 0x03 or 0x04)

Field	Value	Description
Length	[1 byte]	Number of bytes after OpCode
OpCode	0x91	Set Antenna Port
Option	0x03 or 0x04	0x03 indicates only the Power setting will be provided 0x04 indicates Power and Settling time will be provided
TX Logical Antenna	[1 byte]	Specifies the TX antenna the following settings are to be applied.
Read Power	[2 bytes]	The TX read power, in centi-dBm, to be used when this antenna is active for read operations.
Write Power	[2 bytes]	The TX write power, in centi-dBm, to be used when this antenna is active for write operations.
Settling Time	[2 bytes]	The settling time, in microseconds, to be used when this antenna is active. **(Only passed when Option=0x04)**
Note: Repeat the TX Logical Antenna, Power and Settling Time (if Option=0x04) fields for each logical antenna being configured for transmitting.		

Note

When setting parameters for multiple logical antennas all the logical antennas requiring non-default values must be set in one command. If this command is sent twice, only the logical antenna settings in the second invocation will be used.

Note

Commands to set up power and settling time (Option=0x03 and 0x04) do not affect the Set Multi-Antenna Search Configuration.

Set Read TX Power (92h)

The **Set Read TX Power** command sets the default, module-wide, power level to be used for reading tags. The power is specified as a 16-bit value in centi-dBm. For instance, a power of 25 dBm is 2500 centi-dBm, which is 0x09C4.

FF	02	92	09	C4	48	9D
SOH	Length	OpCode	Power in centi-dBm		CRC	

Note

Power settings for individual logical antenna ports specified in Set Antenna's Power and Settling Time override the value set here.

Set Current Tag Protocol (93h)

To select a protocol, send the **Set Current Tag Protocol** code to the reader. A table of valid protocol codes is found in Get Current Tag Protocol (63h). The following example sets the protocol to GEN2:

FF	02	93	00 05	51 7D
SOH	Length	OpCode	Current Protocol	CRC

Only protocols that are enabled in the reader are available. These protocols are listed in the version information for the application, described in Get Boot Loader/Firmware Version (03h).

Note

Changing protocol parameters manually using Set Protocol Configuration (9Bh) will maintain their value across changes in the current protocol using Set Current Tag Protocol (93h). Their values will NOT be reset if the protocol is reset, only if the module is power cycled or the application firmware is reloaded by a Start Bootloader (09h) followed by a Boot Firmware (04h).

Note

Calling Set Current Tag Protocol will reset the Tag Buffer.

Set Write TX Power (94h)

The **Set Write TX Power** command sets the default, module-wide, power level to use for writing to tags. This is necessary because a write operation may require a different RF power setting than a read command. The format and arguments of this command are identical to the **Set Read TX Power** command:

FF	02	94	09 C4	28 5B
SOH	Length	OpCode	Power in centi-dBm	CRC

Note

Power settings for individual logical antenna ports specified in Set Antenna's Power and Settling Time override the value set here.

Set Frequency Hop Table (95h)

The **Set Frequency Hop Table** command sets the list of frequencies and, optionally, the Regulatory hop time, to use when hopping. Each frequency is encoded as a 32-bit value in kHz. For instance, 915MHz is encoded as 915000kHz, which is 0x000DF638. The maximum number of hop frequencies is 62, since that is the maximum number of 32-bit values that can be sent using a message packet. If fewer values are used, only those values are populated in the table and the rest of the slots are ignored.

Note

The data length of this message encodes the number of frequencies to populate into the hop table. The length must be divisible by four for the message to be properly formatted.

Setting Frequencies

This example shows a command that sets up a table with only three values:

FF	0C	95	00	0D	C3	70	00	0D	F6	38	00	0E	26	12	C1	8F
SOH	Length	OpCode	Freq #1				Freq #2				Freq #3				CRC	

The generated hop table has values of 902MHz, 915MHz, and 927.25MHz. For the US region, valid frequencies are 902MHz – 928MHz. If any of the values in the table are invalid, then none of the values are recorded.

The **Set Frequency Hop Table** command should be used for debug only, as there should be no reason to modify the frequency hop table in the field.

C A U T I O N !

Any changes to the frequency hop table could put you out of compliance with the Local Regulatory Requirements (for example, FCC, ETSI, MIC).

Setting Regulatory Hop Time

Adding option field and specifying option=0x01 allows the hop frequency to be customized up to the maximum time allowed by the regulator limits for the region in use.

Set Regulatory Hop Time Command Fields (Option = 0x01)

Field	Value	Description
Length	0x05	Number of bytes after OpCode
OpCode	0x95	Set Frequency Hop Table
Option	0x01	Indicates the Regulatory Hop Time will be passed.
Regulatory Hop Time	[4 bytes]	The time between frequency hops, in milli-seconds, to be used for the defined hop table. Note: The max hop time is restricted by regulatory limits based on the region set via Set Current Region (97h).

Note

Option=0x00 is reserved and not the same as when no option is specified.

Set User GPIO Outputs (96h)

There are two GPIO outputs available for use. The **Set User GPIO Outputs** command has been “overloaded” to create a mechanism to return the current state of both GPIO pins. To set either GPIO Output pin, send the following command. Note that the data length is 2 bytes:

FF	02	96	01	01	28 E0
SOH	Length	OpCode	GPIO #	Output Value	CRC

This example sets the GPIO Output #1 to ‘1’ (high). To get the current status of the GPIO output pins, send the same command, with the length set to 0:

FF	00	96	1D 99
SOH	Length	OpCode	CRC

The response looks similar to the **Get User GPIO Inputs** command:

FF	02	96	00 00	00	01	29 E0
SOH	Length	OpCode	Status	Output #1	Output #2	CRC

Set Current Region (97h)

The **Set Current Region** command sets the current region for use in the reader. The list of region codes are found in Region codes. See Regional Support for more information on regional functionality.

Setting the region performs the following:

1. Frequency hop table is set to the default for the region.
2. Any other region specific settings are set (i.e. LBT) to default values unless otherwise specified. Currently only LBT is configurable for regions supporting it.

The Set Current Region command supports two command syntaxes:

- The **basic syntax** for all regions where only the region code is specified. For example, to set the region to KR:

FF	01	97	03	4B BE
SOH	Length	OpCode	Region	CRC

- The **extended syntax** allowing the region code plus LBT Enabled (if supported by Region) to be set. For example, to set the Region to EU3 and Enable LBT:

FF	02	97	08	01	19 FD
SOH	Length	OpCode	Region	LBT Enable	CRC

Note

When LBT is disabled in the Open region the M5e/M5e-Compact will implement frequency hopping as implemented for the NA region with a frequency hop interval of 400ms.

Set Power Mode (98h)

The **Set Power Mode** command can set the M5e to four different power management modes. See Available power modes.

Use the **Set Power Mode** command to set the power mode to maximum saving mode by sending the following:

FF	01	98	03	44	BE
SOH	Length	OpCode	Power Mode	CRC	

Set User Mode (99h)

Use the **Set User Mode** command to set the user mode to the type of application in which the M5e will be configured for the optimal GEN2 and Search Strategy settings. Setting the User Mode to an unsupported value will not modify the current setting.

Available Gen2 User Modes

Value	User Mode	Session	Target	M	Q Value (Static/ Dynamic)
0x00	NONE (Default)	0	A only	4	InitQ=2 (Dynamic)
0x01	PRINTER	0	A only	4	InitQ=2 (Dynamic)
0x02	Unsupported				
0x03	PORTAL	1	A only	4	InitQ=3 (Dynamic)

Send the following to set the user mode to printer:

FF	01	99	01	45	BC
SOH	Length	OpCode	User Mode	CRC	

Note

Setting the parameters using **Set User Mode** will override the values set by a previous call to Set Protocol Configuration (9Bh).

Note

For details on the behavior of the Gen2 parameter settings see the *EPCGlobal Gen2 RFID Air Interface Specification v1.2.0* or later.

Set Reader Configuration(9Ah)

The **Set Reader Configuration** command is used to set several configuration options on the reader. The Option byte indicates whether to use the new key/value pair setting or the previous, bitwise, settings. One example configuration and the corresponding Options value is specified in the command example below. For other combinations the hex value for Options will need to be calculated by OR'ing the bits for each setting.

Available Configuration Options

Option	Key	Value	Reader Configuration setting
0x00	Indicates the deprecated format of this command will be used: Appendix D: Set Reader Configuration(9Ah)		

0x01	*Use Antenna Port as Unique Identifier of Tag Buffer Entry* = 0x00	0x00	Antenna port is a unique characteristic of a Tag Buffer Entry (*Default*)
		0x01	Antenna port is ignored for Tag Buffer Entries.
	Transmit Mode = 0x01	0x00	High Performance Mode (disables low power mode)
		0x01	Low Power Mode (disables high performance mode)
	Maximum EPC Length = 0x02	0x00	Maximum EPC length of **96 bits** *(Default)*
		0x01	Maximum EPC length of **496 bits**
	Use GPIO as Antenna Switch = 0x03	0x00	Do **NOT** use GPO lines for antenna control (allows for 2 logical antenna ports) *(Default)*
		0x01	Use **GPO 1 f**or antenna control (allows for 4 logical antenna ports)
		0x02	Use **GPO 2** for antenna control (allows for 4 logical antenna ports)
		0x03	Use Both **GPO 1 & 2** for antenna control (allows for 8 logical antenna ports)
		Note: See Using a Multiplexer and Set Antenna Port (91h) for details on setting up logical antenna configurations. Note: When changed any previous settings applied with Set Antenna Port (91h) will be reset to defaults.	
	Check Antenna Connection = 0x04	0x00	**Disable** check for a connected antenna before every transmission. *(Default)*
		0x01	**Enable** check for a connected antenna before every transmission. Note: If enabled, it is recommended Get Antenna Configuration (61h) be used to verify all expected antennas are detectable. If detection fails for an antenna it will be silently skipped during a search.
	Record the highest RSSI seen = 0x06	0x00	The RSSI Tag Read Meta Data value is the value for the last read tag for each tag buffer entry when this is **Disabled**. (*Default*)
		0x01	When **Enabled** the RSSI Tag Read Meta Data value will be the highest value recorded during the Read Tag Multiple (22h) search operation.
	Tag Data as Unique Identifier of Tag Buffer *entry.* = 0x08	0x00	**Tag Data is a unique** characteristic of a Tag Buffer Entry.
		0x01	**Tag Data is ignored** for Tag Buffer Entry uniqueness (*Default*)

Send the following to enable the Low power transmit mode:

FF	03	9A	01	01	01	AE	5C
SOH	Length	OpCode	Option	Key	Value	CRC	

Note

Multiple settings cannot be applied in a single command.

Set Protocol Configuration (9Bh)

The **Set Protocol Configuration** command is used to set protocol-specific configuration parameters. The table below defines the currently available protocol-specific parameters

which can be set and the supported settings. Additional parameters will be supported in the future.

Protocol Value	Parameter	Option	Value
0x05 (Gen2)	**Gen2 Session** used for Inventory commands. = **0x00**	N/A	**0x00** - Session = S0 - *(Default)*
			0x01 - Session = S1
			0x02 - Session = S2
			0x03 - Session = S3
	Gen2 Target used for Inventory commands. = **0x01**	0x01 (Static)	**0x00** - Search Target = A (*Default*)
			0x01 - Search Target = B
		0x00 (Toggle A<>B) Note: Searches using the toggling behavior will use the value Target was in when the last search ended for the start of the next search. To reset the target this command must be reissued.	**0x00** - Search Target starts at A and toggles to B during Read Tag Multiple (22h) after no more tag are found with A.
			0x01 - Search Target starts at B and toggles to A during Read Tag Multiple (22h) after no more tag are found with B.
	Gen2 M Value used for Inventory commands. = **0x02**	N/A	**0x01** (M = 2)
			0x02 (M = 4) (*Default*)
			0x03 (M = 8)
	Gen2 Q Value = **0x12**	0x00 = **Dynamic** - Automatically adjust Q value.	N/A
		0x01 = **Static** - User specified Q Value between 0 and 15.	**[1 byte]** (**0x00-0x0F**)
	Write Response Wait Time used to reduce the time a word write operation will wait for a tag response before moving to next word write. = **0X3F**	**0x00 = Early Exit** - If the tag's response to a word write is detected it moves onto the next word write, otherwise waits for timeout Value.	**[2 bytes]** - Wait timeout in microseconds: • Minimum = 1000us (0x03E8) • Maximum = 21000us (0x5208) Note: Default per Gen2 spec is Early Exit with a 20000us timeout.
		0x01 = Fixed Wait Time - Always waits the specified timeout Value per word write.	

Note

Setting the parameters using **Set Protocol Configuration** will override the implicit value set by a previous call to **Set User Mode**. Likewise, a subsequent call to Set User Mode will override the parameters set by Set Protocol Configuration. **Get Protocol Configuration** will always return the current setting regardless of whether it was set by Set Protocol Configuration or Set User Mode.

Note

Changing these protocol parameters manually using Set Protocol Configuration (9Bh) will maintain their value across changes in the current protocol using Set Current Tag Protocol (93h). Their values will NOT be reset if the protocol is reset, only if the module is power cycled or the application firmware is reloaded by a Start Bootloader (09h) followed by a Boot Firmware (04h).

Examples

Session

The following example sets the Gen2 session to S2:

FF	03	9B	05	00	02	DC	EA
SOH	Length	OpCode	Protocol	Parameter	Value	CRC	

Q Value

The Q valued can be set to a static user specified value or to dynamically change based on ThingMagic internal algorithms. When using the dynamic Q setting the Value parameter is dropped in the command.

The following example statically sets the Q value to 6:

FF	04	9B	05	12	01	06	80	A9
SOH	Length	OpCode	Protocol	Parameter	Option	Value	CRC	

The following example sets dynamic Q value adjusting:

FF	03	9B	05	12	00	CE	E8
SOH	Length	OpCode	Protocol	Parameter	Option	CRC	

Note

For details on the behavior of the Gen2 parameter settings see the *EPCGlobal Gen2 RFID Air Interface Specification v1.2.0* or later.

Get Application Commands

The Get commands listed in the following table are used to get settable parameters from the Mercury Embedded module. All of the commands have a data length of zero, and return data or a fault code.

Applications Commands – Get

OpCode	Command Name	Bootloader	App Firmware
0x10	Get Hardware Version (10h)	Y	Y
0x61	Get Antenna Configuration (61h)	N	Y
0x62	Get Read TX Power (62h)	N	Y
0x63	Get Current Tag Protocol (63h)	N	Y
0x64	Get Write TX Power (64h)	N	Y
0x65	Get Frequency Hop Table (65h)	N	Y
0x66	Get User GPIO Inputs (66h)	N	Y
0x67	Get Current Region (67h)	N	Y
0x68	Get Power Mode (68h)	N	Y
0x69	Get User Mode (69h)	N	Y
0x6A	Get Reader Configuration(6Ah)	N	Y
0x6B	Get Protocol Configuration (6Bh)	N	Y
0x6C	Get Reader Statistics (6Ch)	N	Y
0x70	Get Available Protocols (70h)	N	Y
0x71	Get Available Regions (71h)	N	Y
0x72	Get Current Temperature (72h)	N	Y

Error Status Codes

- FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h
- FAULT_MSG_INVALID_PARAMETER_VALUE - 105h
- FAULT_AHAL_TRANSMITTER_ON – 502h

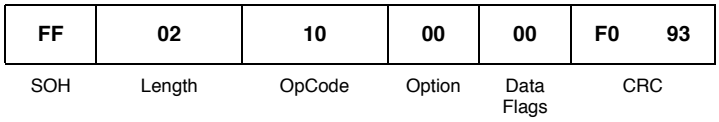
Get Hardware Version (10h)

The Get Hardware Version command is used to get information about the module it is executed on. Most of the information is not currently user-relevant but will often be required by ThingMagic Support to help diagnose problems.

Get Hardware Version Fields

Field	Value	Description
Option	0x00	Currently the only supported value.
[1]Data Flags	0x00	When no flags are set all data will be returned.
	0x40	Returns the Reader Serial Number, as printed on the barcode label, as ASCII values.

The format of this command is:



Note

On many earlier hardware revisions a status code of 0x0109 (FAULT_UNIMPLEMENTED_FEATURE) will be returned. This indicates the necessary information for this command was not stored on the module at that time. It is not cause for concern.

Get Antenna Configuration (61h)

The **Get Antenna Configuration** command returns the current antenna configuration including which antennas are set to transmit and receive, and which ports have antennas attached.

Reliable antenna detection requires that an attached antenna pass at least a small amount of DC current. Many antennas do not pass DC current. Due to such antennas an indication that a port is terminated is always accurate, but an indication that a port is not terminated may not be accurate.

Non-Multiplexer Options

For backward compatibility the following syntax(Option=0x00 and 0x01) is supported when using only the physical module ports directly. For new development the Logical Antenna Options syntax should be used.

The format of the command is:

FF	01	61	00	BD BD
SOH	Length	OpCode	Option	CRC

The reply is formatted similar to the **Set Antenna Port** command. Data byte 01 returns the current TX antenna, and data byte 02 returns the current RX antenna. The default antenna configuration is TX on port #1, and RX on port #2, except on the M5e-Compact which only has 1 port so the M5e-Compact's default is TX on port #1 and RX on port #1.

FF	02	61	00 00	01	02	4E 21
SOH	Length	OpCode	Status	TX Ant Num	RX Ant Num	CRC

The **Get Antenna Configuration** command also includes an option that also returns information on which port(s) the antenna is connected.

For example, a command is sent:

FF	01	61	01	BD BC
SOH	Length	OpCode	Option	CRC

The response returns the ports for TX and RX and also detects whether an antenna is connected to the port:

FF	04	61	00 00	01	01	00	01	A7 02
SOH	Length	OpCode	Status	TX Ant Num	RX Ant Num	Port 1 Not Connected	Port2 Connected	CRC

Logical Antenna Options

The new Get Antenna Configuration options allow you to get the logical antenna configuration of the module. This syntax supports detecting antennas connected to a multiplexer provided the appropriate GPIO control lines have been enabled using Set Reader Configuration(9Ah).

Get Antenna Port Command Fields

Field	Value	Description
Length	0x01	Number of bytes after OpCode
OpCode	0x61	Get Antenna Port
Option	0x02	Return the Antenna Search Order as specified by Set Multi-Antenna Search Configuration
	0x03	Returns all TX Antennas and their associated Power settings as specified by Set Antenna's Power and Settling Time
	0x04	Returns all TX Antennas and their associated Power and Settling time settings as specified by Set Antenna's Power and Settling Time
	0x05	Returns all "valid" Logical Antenna Ports and their connection status.
	Note: "Valid" ports are defined by the tables for the specific control lines in use.	

For options 0x02, 0x03 and 0x04 the returns values, fields and order are equivalent to the settings using the same option with Set Antenna Port (91h).

Get Antenna Port Status Response Fields (Option = 0x05)

Field	Value	Description
Length	[1 byte]	Number of bytes after OpCode
OpCode	0x91	Set Antenna Port
Status	[2 bytes]	Status of command
Option	0x05	Indicates the data to follow contains each logical antenna available and its connection status (is an antenna detected).
Logical Antenna	[1 byte]	Specifies the logical antenna the following connection status applies to.
Connection Status	0x00	No antenna detected
	0x01	Antenna (or termination) detected.

Field	Value	Description
Repeat for all available logical antennas, as defined by the settings in Set Reader Configuration(9Ah)		

Get Read TX Power (62h)

For deprecated version of this command see Appendix D: Get Read TX Power (62h)

The **Get Read TX Power** command returns the current TX power and, optionally, the minimum and maximum power levels supported by the module (and region setting) for reading tags, in centi-dBm.

The option field allows you to specify what data you want returned:

Option	Data returned
0x00	Returns only the current TX power.
0x01	Returns the current TX power and the maximum and minimum power levels for this module.

The following example gets the full set of power data:

FF	01	62	01	BE BC
SOH	Length	OpCode	Option	CRC

The example response contains the option specified followed by 2 byte fields for each power value:

current = 0x0BB8 = 3000 centi-dBm = 30.00 dBm
max = 0x0BB8 = 3000 centi-dBm = 30. 00 dBm
min. = 0x01F4 = 500 centi-dBm = 5.00 dBm

FF	07	62	00 00	01	0B B8	0B B8	01 F4	1F B7
SOH	Length	OpCode	Status	Option	Current centi-dBm	Max centi-dBm	Min centi-dBm	CRC

Get Current Tag Protocol (63h)

The **Get Current Tag Protocol** command returns the currently set tag protocol(s). This is the protocol being used by the reader. The following table assigns a 16-bit code to each available protocol. This table will be updated as new protocols are added:

Protocol Name	16-bit Code
None Selected	0x0000
EPC0 / EPC0+ Matrics	0x0001
EPC1	0x0002
ISO 18000-6B	0x0003
EPC0+ Impinj	0x0004
GEN2	0x0005
UCODE	0x0006

The command format is shown in the following example:

FF	00	63	1D 6C
SOH	Length	OpCode	CRC

Note

The M5e and M5e-Compact only support GEN2

The reply is similar to the **Set Current Tag Protocol** command. Only one protocol can be set at a time:

FF	02	63	00 00	00 01	21 42
SOH	Length	OpCode	Status	Current Protocol	CRC

Get Write TX Power (64h)

For deprecated version of this command see Appendix D: Get Write TX Power (64h)

The **Get Write TX Power** command returns the current TX power and, optionally, the minimum and maximum power levels supported by the module for reading tags, in centi-dBm.

The option field allows you to specify what data you want returned:

Option	Data returned
0x00	Returns only the current TX power.
0x01	Returns the current TX power and the maximum and minimum power levels for this module.

The following example gets the full set of power data:

FF	01	64	01	B8 BC
SOH	Length	OpCode	Option	CRC

The example response contains the option specified followed by 2 byte fields for each power value:

current = 0x0BB8 = 3000 centi-dBm = 30.00 dBm
max = 0x0BB8 = 3000 centi-dBm = 30.00 dBm
min. = 0x01F4 = 500centi-dBm = 5.00 dBm

FF	07	64	00 00	01	0B B8	0B B8	01 F4	FF BC
SOH	Length	OpCode	Status	Option	Default centi-dBm	Max centi-dBm	Min centi-dBm	CRC

Get Frequency Hop Table (65h)

The **Get Frequency Hop Table** command gets the current frequency hop table and, optionally, the frequency hop time, used.

Get Frequencies

The following command gets the list of frequencies current used in the hop table.

FF	00	65	1D	6A
SOH	Length	OpCode	CRC	

The reply format is similar to the **Set Frequency Hop Table** command. The length must be set to a multiple of four, and a maximum of 62 hop frequencies is returned. The frequencies are returned in kHz. Thus, frequency #1 (0xDC370) corresponds to 902,000 kHz:

FF	0C	65	00	00	00	0D	C3	70	00	0D	F6	38	00	0E	26	12	60	B2
SOH	Length	OpCode	Status		Freq #1				Freq #2				Freq #3				CRC	

Get Regulatory Hop Time

Adding the option field and specifying option=0x01 returns the hop frequency in use.

Get Regulatory Hop Time Command Fields (Option = 0x01)

Field	Value	Description
Length	0x01	Number of bytes after OpCode
OpCode	0x65	Get Frequency Hop Table
Option	0x01	Indicates the Regulatory Hop Time will be passed.

Get Regulatory Hop Time Response Fields (Option = 0x01)

Field	Value	Description
Length	0x05	Number of bytes after OpCode
OpCode	0x65	Get Frequency Hop Table
Status	[2 bytes]	Status
Option	0x01	Indicates the Regulatory Hop Time will be passed.
Regulatory Hop Time	[4 bytes]	The hopping interval, in milli-seconds, currently used for the defined hop table.

Note

Option=0x00 is reserved and not the same as when no option is specified.

Get User GPIO Inputs (66h)

The **Get User GPIO Inputs** command returns the status of the User Input GPIO lines:

FF	00	66	1D	69
SOH	Length	OpCode	CRC	

The status of the GPIO lines is returned.

FF	02	66	00	00	00	01	CA	B2
SOH	Length	OpCode	Status		Input #1	Input #2	CRC	

Note

The GPI lines are level sensed and not edge sensed. When Get User GPIO Inputs is called it will respond with the signal level (high or low) on the line at that moment.

Get Current Region (67h)

The **Get Current Region** command gets the current regulatory region set for the reader. Currently available region codes are shown in the following table. Details on the Regulatory Compliance for each region can be found in Regional Support:

Region codes

Region	Code
NA	0x01
EU	0x02
KR	0x03
IN	0x04
PRC	0x06
EU2	0X07
EU3	0x08
KR2	0x09
AU	0x0B
NZ	0x0C
Open	0xFF

To get the current region send the following command:

FF	00	67	1D 68
SOH	Length	OpCode	CRC

The command generates the following reply, which indicates that the reader is set to the NA region:

FF	01	67	00 00	01	B4 80
SOH	Length	OpCode	Status	Region	CRC

Get Power Mode (68h)

The **Get Power Mode** command returns the current Power Mode of the unit. The values for the available Power Modes are shown in the table.

Available power modes

Hex	Power Mode
0x00	Full power mode
0x01	Minimal saving mode.
0x02	Medium saving mode.
0x03	Maximum saving mode.
0x04-0xFF	Reserved for future use.

Note

Maximum Saving Mode only supports communications at 9600 baud

The command gets the current power mode setting from the module:

FF	00	68	1D 67
SOH	Length	OpCode	CRC

The example reply shows that the system is in the medium power saving mode:

FF	01	68	00 00	02	A4 BD
SOH	Length	OpCode	Status	Power Mode	CRC

Get User Mode (69h)

The **Get User Mode** command returns the type of application in which the M5e/M5e-Compact can be used. The list of user modes and their corresponding protocol configuration settings can be found in Available Gen2 User Modes.

The command gets the current user mode setting from the module:

FF	00	69	1D 66
SOH	Length	OpCode	CRC

The example reply shows the current user mode setting:

FF	01	69	00 00	01	97 8F
SOH	Length	OpCode	Status	User Mode	CRC

Get Reader Configuration(6Ah)

For the previous version of this command see Appendix D: Get Reader Configuration(6Ah)

The **Get Reader Configuration** command returns the current reader configuration for the setting specified by the key field as defined in Set Reader Configuration(9Ah) when using Option=0x01. When using Option=0x00 it will use the previous format of the command.

Send the following command to get the current Unique Read setting:

FF	02	6A	01	00	2E 4E
SOH	Length	OpCode	Option	Key	CRC

The example reply shows the current configuration setting:

FF	03	6A	00 00	01	00	01	AF 5C
SOH	Length	OpCode	Status	Option	Key	Value	CRC

Get Protocol Configuration (6Bh)

The **Get Protocol Configuration** command is used to get current protocol-specific configuration parameters settings as specified in Set Protocol Configuration (9Bh).

Send the following command to get the current Gen2 Session configuration parameter value:

FF	02	6B	05	00	3A 6F
SOH	Length	OpCode	Protocol	Parameter	CRC

The example reply shows the Gen2 session is set to S2 (0x02):

FF	03	6B	00	00	05	00	02	08	76
SOH	Length	OpCode	Status		Protocol	Parameter	Value	CRC	

Get Reader Statistics (6Ch)

The Get Reader Statistics command allows the user to get and reset various statistics on the reader operation. The command takes the following fields:

Get Reader Statistics Request Fields

Field	Value
Option (1 byte)	• **0x00** - Get statistics specified by the Statistics Flag • **0x01** - Reset the specified statistic. • **0x02** - Get requested statistics for each port and include port ID in response. • **0x03** - Get requested statistics for Request Ports.
Statistics Flag (1 byte)	Each bit corresponds to a specific statistic to be returned. See the Available Statistics table for bit values.
Requested Ports Length (1 byte)	Note: Only include if Option = 0x03 Number of ports statistics are requested for. Indicates the number (N) bytes to follow.
Requested Ports (N bytes)	Note: Only include if Option = 0x03 Port ID of ports to return statistics for.
Note: When getting statistics for all ports the returned ports is impacted by the Available Configuration Options I Check Antenna Connection. • If checking is enabled then only values for connected/detectable antenna ports are returned. • If checking is disabled then values for all antenna ports are returned. However, if a port is not connected it is possible this command will fail with a FAULT_HIGH_RETURN_LOSS – 505h status.	

Available Statistics

Statistics Flag Bit	Statistic
Bit 0 (0x01)	**RF On Time (ms)**- Indicates the aggregate time the transmitter has been on, in milli-seconds, since the counter was last reset. **Returned value** contains N bytes (N/2 on the M5e-Compact), Where N = the number of valid logical antennas based on the Antenna Ports configuration X 4 bytes for each antenna. Note: Clock rolls over every 2^{32} milliseconds and must be taken into account by user application.
Bit 1 (0x02)	**Noise Floor** - When requested the reader will listen on each available logical antenna and report the noise floor in units equivalent to the units returned for Tag Read Meta Data RSSI value. **Returned value** contains N bytes (N/2 on the M5e-Compact), Where N = the number of valid logical antennas based on the Antenna Ports configuration x 1 byte for each antenna, in ascending antenna order. Note: A single RF operation (read, etc.) must be performed before this operation can return valid data.
Bit 3 (0x08)	**Noise Floor with TX On** - When requested the reader will transmit a CW signal on each available logical antenna and report the noise floor detected in units equivalent to the units returned for Tag Read Meta Data RSSI value. **Returned value** contains N bytes (N/2 on the M5e-Compact), Where N = the number of valid logical antennas based on the Antenna Ports configuration x 1 byte for each antenna, in ascending antenna order.
Note: Multiple statistics can be requested by a single command by performing a binary OR (setting each desired Statistics Flag bit to 1) on the desired flags and sending the result as the Statistics Flag byte.	

The response to a Get Reader Statistics command contains the following information:

Get Statistics Response Fields

Field	Description
Status (2 bytes)	Standard response status
Option (1 byte)	Same as the Requested Value
Requested Statistics Flag (1 byte)	Same as the Requested Value
For each individual statistics requested as part of the Requested Statistics Flag the following fields will be repeated according to the Statistics Response value. They will be in bit value order.	
Statistic Flag (1 byte)	The bit corresponding to the requested statistic.
Data Length (1 byte)	Indicates how many bytes are in this statistics response value.
Antenna Port ID	Note: Only included when Option=0x02 Antenna port ID[1] of the following Statistic value.
Statistics Response Value (N bytes as indicated by Data Length)	The response value of the statistic as specified in Available Statistics
Note: 1 - When Using a Multiplexer all antenna values correspond to the Logical Antenna Setting antenna value.	

To request the Noise floor with TX on, including portID for all ports, send the following command:

FF	02	6C	02	08	XX XX
SOH	Length	OpCode	Option	Statistics Flag	CRC

The response (from an M5e with two antennas), containing the RF On Time for each antenna would look like:

FF	08	6C	00 00	02	08	08	04	01	44	02	25	XX XX
SOH	Length	OpCode	Status	Option	Requested Stats Flag	Stat Flag (Noise)	Data Length	Antenna ID	Noise floor	Antenna ID	Noise floor	CRC

To reset the RF On Time statistic send the following command:

FF	02	6C	01	01	4E 89
SOH	Length	OpCode	Option	Statistics Flag	CRC

Get Available Protocols (70h)

The **Get Available Protocols** command returns the list of protocols that the reader is capable of reading. The protocol codes are defined in Get Current Tag Protocol (63h). The total number of protocols in the system cannot exceed 32. The command is generated as follows:

FF	00	70	1D 7F
SOH	Length	OpCode	CRC

The following Microprocessor response shows that the reader is capable of reading one protocol – GEN2:

FF	02	70	00 00	00 05	3B 75
SOH	Length	OpCode	Status	Protocol #1	CRC

Get Available Regions (71h)

The **Get Available Regions** command returns the list of regions that the reader is capable of working in. The total number of regions in the system cannot exceed 248. The list of region codes are found in Region codes. The command is generated as follows:

FF	00	71	1D 7E
SOH	Length	OpCode	CRC

The response shows that this reader is capable of reading in four regions – NA, EU, KR, and EU2:

FF	03	71	00 00	01	02	03	07	39 FF
SOH	Length	OpCode	Status	Region#1	Region#2	Region#3	Region#7	CRC

Get Current Temperature (72h)

The **Get Current Temperature** command returns the current board component temperature on the module in degrees Celsius as a signed 8-bit integer. This information can be used to determine if the module is exceeding its recommended operating temperature range. If the returned value exceeds the values specified in the table below an effort should be made to reduce the ambient temperature of the module or reduce the module's duty cycle.

Module	Warning Temperature
M5e	85°C
M5e-C	80°C

Note

In M5e Firmware release v1.5.1 the temperature threshold value changed. This change is due to a value calculation change and not a change in the actual temperature threshold. The previous values returned were in uncalibrated "units", not actual degress Celcius.

The command is generated as follows:

FF	00	72	1D 7D
SOH	Length	OpCode	CRC

The response returns a signed 8-bit integer in degrees Celsius, 0x24 = 36°C

FF	01	72	00 00	24	48 23
SOH	Length	OpCode	Status	Temperature in Celsius	CRC

FCC Test Commands

The following OpCodes are used for test purposes including regulatory certification testing.

Note

Set Power Mode to Full 0x00 before running FCC commands.

FCC Test Commands

OpCode	Command Name	Arguments	Return	BL	App	M5e/M5e-Compact
0xC1	Set Operating Frequency (C1h)	Frequency in kHz	none	N	Y	Y
0xC3	Transmit CW Signal (C3h)	CW mode, [timeout]	none	N	Y	Y

Set Operating Frequency (C1h)

The **Set Operating Frequency** command takes a 32-bit frequency value, expressed in kHz. For instance, to set the frequency to 915.26MHz, send the data value 915260 (0x000D F73C) to the reader.

FF	04	C1	00	0D	F7	3C	F3	B7
SOH	Length	OpCode	Freq to Set				CRC	

Transmit CW Signal (C3h)

The **Transmit CW Signal** command turns the Continuous Wave (CW) signal On or Off or enables a PRBS signal. Sending 0x00 turns Off the CW signal; sending 0x01 turns On the CW signal; Sending 0x02 turns on a PRBS signal. The CW signal is transmitted at the last used power level.

This example shuts off CW:

FF	01	C3	00	1F	BD
SOH	Length	OpCode	CW	CRC	

Some regulatory testing requires a PRBS (Pseudo-Random Bit Sequence) signal to simulate data transmission. To send a PRBS signal, along with sending CW=0x02, the Length field must be changed to 0x03. This indicates PRBS will be used and an extra 2 byte Timeout field must be added. When this PRBS mode is used the CW signal is on until the timeout expires, during that period the reader does not respond to commands.

The following example generates a PRBS CW signal for 256 ms, 100% duty cycle:

FF	03	C3	02	01	00	01	00
SOH	Length	OpCode	CW/PRBS	Timeout		CRC	

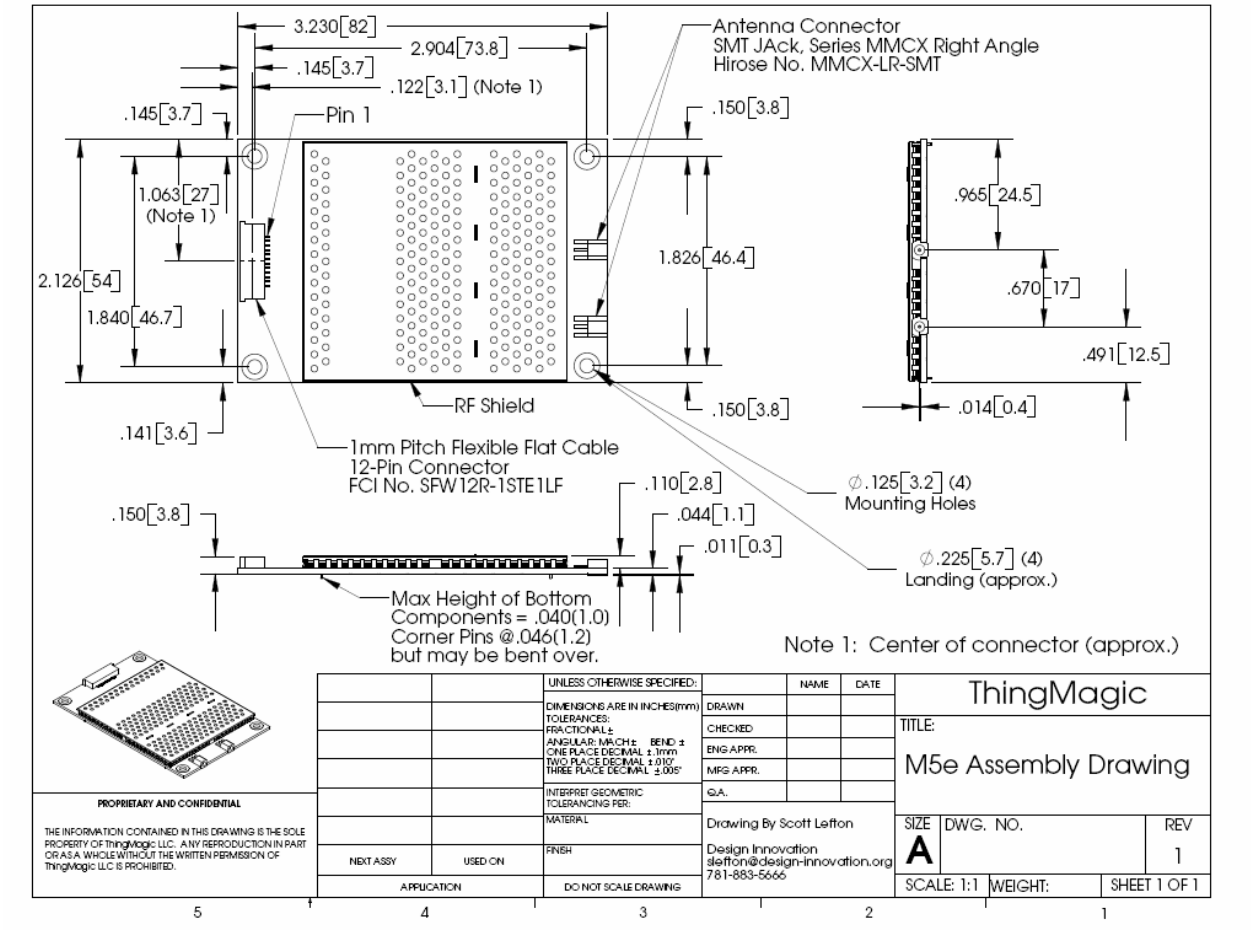
Appendix A: Hardware Details

This Appendix details the mechanicals for the embedded modules and also provides pin 1 locations for the M5e and M5e-Compact serial connectors.

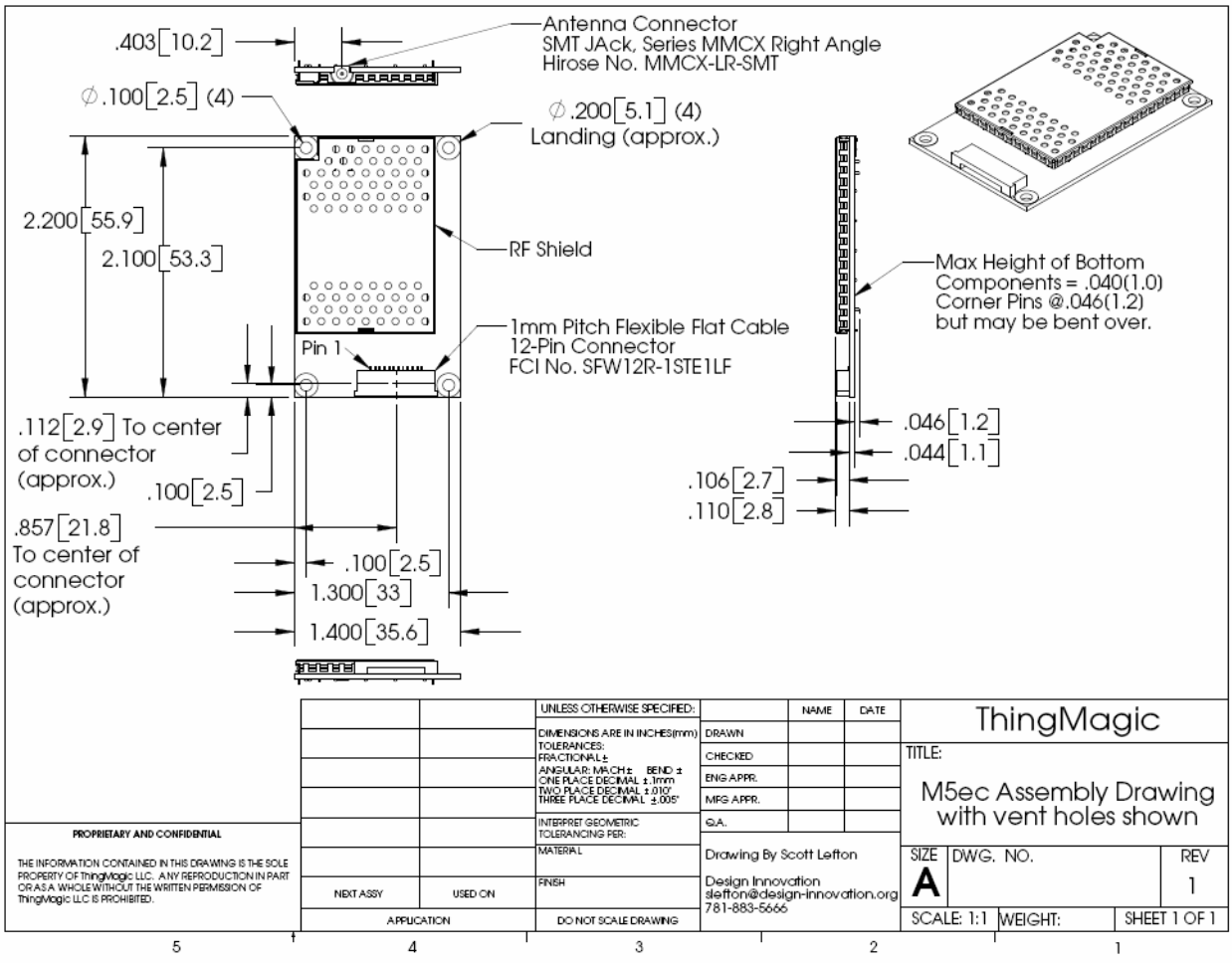
Mechanicals

The following figures detail the hardware layouts that comprise the Mercury Embedded Modules.

M5e Mechanicals



M5e-Compact Mechanical



Antenna Connector

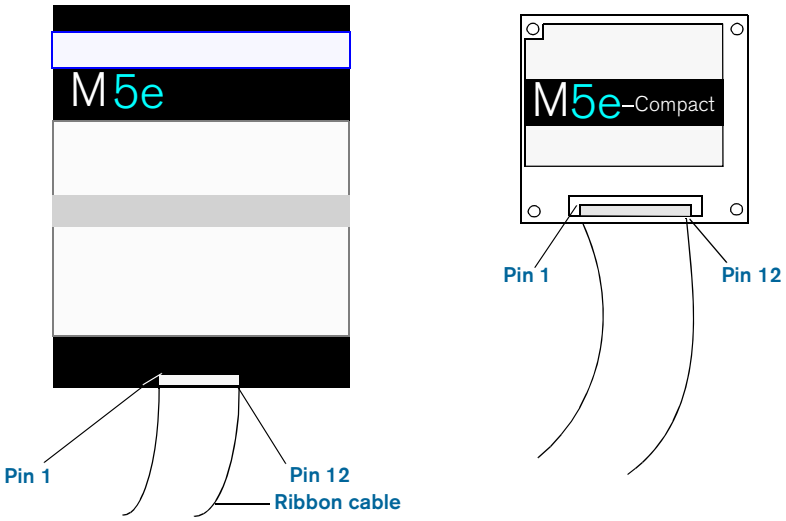
The M5e has two MMCX connectors and the M5e-Compact has one MMCX connector for interfacing to the antennas.

Communications Connector

The communications interface on the modules provides power, serial communications signals, and access to the GPIO inputs and outputs.

The M5e and M5e-Compact have a 12-pin connector. For the interface pin-out, see M5e/ M5e-Compact Digital Connectors.

The following figure shows the diagrams of the M5e, and M5e-Compact communications interface as you face the boards.



Note

The flat ribbon cable that connects with the communications interface on the M5e and M5e-Compact embedded modules is available from Parlex, Part Number: 100R12-152B; and can be purchased from standard electronic component suppliers.

Appendix B: Getting Started - Devkit

Devkit Hardware

Included Components

With the devkit, you will receive the following components:

- The M5e/C module and power/interface developers board
- One USB cable (may contain two host-side connectors, do not use the **Red** one)
- One antenna
- One coax cable
- One 9V power supply
- International power adapter kit
- Sample tags
- Two paper inserts:
  - *QuickStart Guide* - Details on which documents and software to download to get up and running quickly, and how to register for and contact support..

Setting up the DevKit

When setting up the DevKit, use the following procedures:

- Connecting the Antenna
- Powering up and Connecting to a PC

Connecting the Antenna

ThingMagic supplies one antenna that can read tags from 20' away with most of the provided tags. The antenna is monstatic. Use the following procedure to connect the antenna to the DevKit.

1. Connect one end of the coax cable to the antenna.
2. Connect the other end of the cable to Ant 1 connector on the DevKit.

Powering up and Connecting to a PC

After connecting the antenna you can power up the DevKit and establish a host connection.

1. Connect the USB cable (use only the black connector) from a PC to the developer's kit. There are two Devkit USB Interface options, use only the USB/RS232 connector.
2. Plug the power supply into the DevKit's DC power input connector.
3. The LED next to the DC input jack, labeled DS1, should light up. If it doesn't light up check jumper J17 to make sure the jumper is connecting pins 2 and 3
4. Follow the steps in Devkit USB Interface and make note of the COM port or /dev device file, as appropriate for your operating system the USB interface is assigned.
5. To start reading tags start the Demo Application.

W A R N I N G !

While the module is powered up, do not touch components. Doing so may be damaged the devkit and M5e module.

Devkit USB Interface

USB/RS232

The USB interface (connector labeled **`USB/RS232`**) closest to the power plug is to the RS232 interface of the M5e through an FTDI USB to serial converter. The drivers for it are available at

http://www.ftdichip.com/Drivers/VCP.htm

Please follow the instructions in the installation guide appropriate for your operating system.

Native USB

Not applicable to the M5e.

Devkit Jumpers

J8

Jumpers to connect M5e I/O lines to devkit.

J9

Header for alternate power supply. Make sure DC plug (J1) is not connected if using J9.

J10, J11, J13, J15

Do not change.

Jump pins `OUT` to `GPIO#` to connect M5e GPIO lines to output LEDs. Jump pins `IN` to `GPIO#` to connect M5e GPIO to corresponding input switches `SW[3-6]GPIO#`. The M5e GPIO lines are not configurable as Inputs or Outputs so the jumpers should not be changed from their factory settings, as follows:

- M5e GPO1 is devkit GPIO1 - J10 jumper pins 1 to 2
  - DS2 lights when GPO1 is low
- M5e GPO2 is devkit GPIO2 - J11 jumper pins 1 to 2
  - DS3 lights when GPO2 is low
- M5e GPI1 is devkit GPIO3 - J13 jumper pins 2 to 3
  - SW6 controls logic level to GPI1
- M5e GPI2 is devkit GPIO4 - J15 jumper pins 2 to 3
  - SW7 controls logic level to GPI2

J14

Can be used to connect GPIO lines to external circuits. If used jumpers should be removed from J10, J11, J13, J15.

J16

Jump pins 1 and 2 or 2 and 3 to reset devkit power supply. Same as using switch SW1 except allows for control by external circuit.

J17

Jump pins 1 and 2 to use the `5V INPUT` and `GND` inputs to provide power. Jump pins 2 and 3 to use the DevKit's DC power jack and power brick power.

J19

Do not change

Devkit Schematics

Available upon request from support@thingmagic.com.

Demo Application

A demo application which supports multi-protocol reading and writing is provided in the MercuryAPI SDK package. The executable for this example is included in the MercuryAPI SDK package under /cs/samples/exe/Universal-Reader-Assistant.exe and the source code under /cs/samples/Universal-Reader-Assistant/Universal-Reader-Assistant.exe.

See the Readme.txt in /cs/samples/Universal-Reader-Assistant/Universal-Reader-Assistant for usage details.

See the *MercuryAPI Programming Guide* for details on using the MercuryAPI.

Arbser Command Line Utility

The ArbSer program is a simple terminal program with which you can communicate with the M5e, and M5e-C modules. It provides several pre-formed commands, as well as a raw message interface that can be used to generate any command. The source code is part of the developer's kit to provide an example of the message format and CRC calculation.

The executable was built using a Windows 2000 PC. If the host PC's operating system is different, the executable should be rebuilt using Microsoft Visual C++ 6.0, or other compatible compiler. If building ArbSer on another platform, some work may be necessary to integrate the serial port properly.

ArbSer provides a help message if it is called with no argument:

C:\> ArbSer

ArbSer Version 3.1.1 compiled on Nov 8 2004 at 11:44:05.

Usage: **ArbSer** <baud> <COM port> <option>

Default baud rate is 9600. To use 115200: ArbSer 115200 <options>

Default COM port is COM1. To use COM3: **ArbSer** <baud> *-c3* <options>

Only one option can be used at a time:

-eraseApp – Erase application FW only (sectors 1-8).

-genMsg – Generates a msg and prints it to console.

-go – Start the application FW.

-l5a MercuryE – Load the M5e or M5e-C application FW(*.sim file) into FLASH

-msg xx xx xx ... – Send properly formatted msg (appends SOH and CRC).

-raw xx xx xx ... – Send raw hexadecimal msg (illegal msgs possible).

-ver – Returns the version information.

When using ArbSer, the serial port and baud rate can be changed from the default of COM1 and 9600 bps. Then, one of the options must be selected. The most useful ones are '*-go*', '*-msg*', and '*-ver*'. These commands provide the ability to exercise every feature of the modules.

Reading a Tag

Now that the module is connected to the PC and is powered on, the ArbSer.exe program is used to communicate with it. Send the sequence of commands described in this section.

The raw hexadecimal output sent through the serial port is shown following the ArbSer command. This is an example of the exact serial traffic sent to the Microprocessor.

Get Version Command

When the unit is first powered up, the boot loader is running. Verify that the module is alive by sending a **Get Version** command:

C:\> **ArbSer** *–ver*

< Equivalent to: FF 00 03 1D 0c >

Valid message received:

Data Length = 14

OpCode = 03

Status = 00 00

Data[000] = 03

Data[001] = 01

Data[002] = 00

Data[003] = 05

Data[004] = FF

Data[005] = FF

Data[006] = FF

Data[007] = FF

Data[008] = 20

Data[009] = 04

Data[010] = 11

Data[011] = 03

Data[012] = 03

Data[013] = 01

Data[014] = 00

Data[015] = 06

Data[016] = 00

Data[017] = 00

Data[018] = 00

Data[019] = 07

CRC = 42EA

The module returns its version information. In the previous example, the version number shows the following:

- Boot loader version #3.1.5
- Application FW build date: 11/03/2004
- Application FW version #3.1.6
- 3 protocols enabled (EPC0, EPC1, ISO18000-6B).

Boot Firmware Command

Next, send a **Boot Firmware** command:

C:\> **ArbSer** *–go*

< Equivalent to: FF 00 04 1D 0B >

Valid message received:

Data Length = 14

OpCode = 04

Status = 00 00

Data[000] = 03

Data[001] = 01

Data[002] = 00

Data[003] = 05

Data[004] = FF

Data[005] = FF

Data[006] = FF

Data[007] = FF

Data[008] = 20

Data[009] = 04

Data[010] = 11

Data[011] = 03

Data[012] = 03

Data[013] = 01

Data[014] = 00

Data[015] = 06

Data[016] = 00

Data[017] = 00

Data[018] = 00

Data[019] = 07

CRC = 4B6A

This returns the same version information as the `-ver` command previously documented. Now that the application FW has been started, the reader is ready to accept protocol commands. To read an EPC0 tag, place the tag about one foot away from the antenna. Then send the following series of commands.

Set Current Region Command

Before you can start reading tags with the M5e and M5e-C, you must set the region in which the reader resides.

For example, if the region is the European Union, using the **Set Current Region** command, set the region as follows:

C:\> **ArbSer** –msg 01 97 02

<Equivalent to: FF 01 97 02 4B BE>

Valid message received:

Data Length = 00

OpCode = 93

Status = 00 00

CRC = 371A

Set Current Tag Protocol Command

When the application FW first starts, there is no default protocol loaded. Thus, the protocol to use must be specified.

For this example, the protocol is set to EPC0 using the **Set Current Tag Protocol** command:

C:\> **ArbSer** –*msg* 02 93 00 01

< Equivalent to: FF 02 93 00 01 51 79 >

Valid message received:

Data Length = 00

OpCode = 93

Status = 00 00

CRC = 371A

Set Read TX Power Command

The default read power setting for the modules is 26.5 dBm. This should be adequate to read the tag from 1 foot. If a different power setting is desired, it should be entered now. The following **Set Read TX Power** command sets the power level to 25.0 dBm (0x09C4):

C:\> **ArbSer** –*msg* 02 92 09 C4

< Equivalent to: FF 02 92 09 C4 FC E5 >

Valid message received:

Data Length = 00

OpCode = 92

Status = 00 00

CRC = 273B

Set Antenna Port Command

The default antenna port configuration is a 2-port antenna, with TX on port 1 and RX on port 2. If a different configuration is being used, it should be changed before attempting to

read tags. For example, to use a one-port configuration that uses only port 1, use the **Set Antenna Port** command:

C:\> **ArbSer** *–msg* 02 91 01 01

< Equivalent to: FF 02 91 01 01 70 3B >

Valid message received:

Data Length = 00

OpCode = 91

Status = 00 00

CRC = 1758

Finally, the module is ready to execute a read command. The timeout for the read can be set anywhere from 0 ms to 65,536 ms. Using the **Read Single Tag ID** command, the timeout is set to 1 s, which is 1000 ms (0x03E8):

C:\> **ArbSer** *–msg* 02 21 03 E8

< Equivalent to: FF 02 21 03 E8 D5 09 >

Valid message received:

Data Length = 0E

OpCode = 21

Status = 00 00

Data[000] = 12

Data[001] = 34

Data[002] = 56

Data[003] = 78

Data[004] = 9A

Data[005] = BC

Data[006] = DE

Data[007] = F0

Data[008] = AA

Data[009] = BB

Data[010] = CC

Data[011] = DD

Data[012] = 23

Data[013] = 79

CRC = 2384

The module should now return the tag ID of the tag. In this example, the 96-bit EPC0 tag ID is (0x12 34 56 78 9A BC DE F0 AA BB CC DD) with a tag ID CRC of 0x2379.

Unexpected Results

Sometimes you do not get the results that you expected. The following sections explain some problems that can occur when the previously listed sequence of commands are used. The following sections explain some of the frequently encountered errors.

Serial Communication Does Not Work

If serial communications fails to work using the **ArbSer** command, check the following:

- The baud rate is correct.
- If the baud rate of the boot loader or application FW was changed, then specify the current baud rate using ArbSer. For example, “ArbSer <baudrate> ...”.
- The correct COM port is being used.
- ArbSer does not return an error if it is able to open the specified COM port. Currently, ArbSer works on COM1 to COM9.
- The commands are properly formed.
- If no SOF byte (0xFF) is used, or an incorrect CRC or number of data elements is specified, then the Microprocessor does not respond to the message at all.
- The module is properly connected and powered on.
- The serial cable should not be a NULL modem cable.
- The PC serial port is working properly.

  An RS-232 line checker is helpful for this.

A simple way to check that serial communications is working is to try sending a **Get Version** command. If a valid response is received, then the physical communication layer is intact and working properly. That is to say that the baud rate, COM port, and cables are all fine.

Commands Return a Non-Zero Status Code

If a command returns a non-zero status code, this is not always an error. However, if a non-zero code is received in response to a **Set** command, such as **Set Current Tag Protocol**, **Set Read TX Power**, or **Set Antenna Port** commands, it is likely that one of the following problems exists:

- The module is still executing the boot loader program.

Send a **Boot Firmware** command to verify that the application FW is running. If the boot loader is currently running, then a set command will return a 0x0101 status code, indicating an invalid OpCode.

- The parameter(s) to the set command are invalid. If an invalid parameter is sent, then a 0x0105 status code is returned, indicating an invalid parameter value.
- If the wrong number of data elements is used, then a 0x0100 status code (wrong number of data) is returned. This could be the result of an ill-formed command, or accidentally using the wrong OpCode.

No Tag ID is Returned

If the **Read Tag ID Single** command returns an error code of 0x0400 (no tag found), then most likely it is a set up related issue. Check the following items:

- The antenna is properly connected to the unit and configured using the **Set Antenna Port** command.
- The tag is the right protocol (that is EPC0) and that the protocol has been set using the **Set Current Tag Protocol** command.
- Try using a longer timeout, such as 30 seconds, and varying the tag's location in relation to the antenna.
- It is also possible the tag may be erased or damaged. Without a second "known good" reader, it is difficult to tell if this is the case. If a "known good" tag is available, try to read it.

Minimum Set of Serial Commands

The minimum set of commands required to setup the M5e for reading are:

Boot Firmware (04h):

```
0xFF 0x00 0x04 0x1D 0x0B
```

Set Current Tag Protocol (93h) [to Gen2]

```
0xFF 0x02 0x93 0x00 0x05 0x51 0x7D
```

Set Current Region (97h) [to EU3]

```
0xFF 0x01 0x87 0x08 0x4B 0xB5
```

Notice you must first boot the application firmware (0x04) so you can successfully execute the setup commands. The "Set" commands cannot be executed when running in the bootloader mode.

Once configure you can perform inventory operations using the Read Tag Multiple (0x22) command. This results in tags being stored in the tag Buffer on the module. You then need to execute Get Tag Buffer (0x29) operations to retrieve the tags. See the docs for more details on how those work. A very simple usages is:

Read Tag Multiple (22h)

```
0xFF 0x02 0x22 0x01 0xF4 0xE7 0x76
```

Get Tag Buffer (29h) - Can return max 13 tags so if Read Tag Multiple resulted in more than 13 being stored then multiple Get Tag Buffer commands must be sent.

```
0xFF 0x03 0x29 0x00 0x00 0x00 0xF4 0x22
```

Notice on Restricted Use of the DevKit

The Mercury5e Developers Kit (DevKit) is intended for use solely by professional engineers for the purpose of evaluating the feasibility of applications.

The user's evaluation must be limited to use within a laboratory setting. This DevKit has not been certified for use by the FCC in accordance with Part 15 of the FCC regulations, ETSI, KCC or any other regulatory bodies and may not be sold or given for public use.

Distribution and sale of the DevKit is intended solely for use in future development of devices which may be subject to regional regulatory authorities governing radio emission. This DevKit may not be resold by users for any purpose. Accordingly, operation of the DevKit in the development of future devices is deemed within the discretion of the user and the user shall have all responsibility for any compliance with any regional regulatory authority governing radio emission of such development or use, including without limitation reducing electrical interference to legally acceptable levels. All products developed by user must be approved by the appropriate regional regulatory authority governing radio emission prior to marketing or sale of such products and user bears all responsibility for obtaining the prior appropriate regulatory approval, or approval as needed from any other authority governing radio emission.

Appendix C: Error Messages

Common Error Messages

The following table lists the common faults discussed in this section.

Fault Message	Code
FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h	100h
FAULT_INVALID_OPCODE – 101h	101h
FAULT_UNIMPLEMENTED_OPCODE – 102h	102h
FAULT_MSG_POWER_TOO_HIGH – 103h	103h
FAULT_MSG_INVALID_FREQ_RECEIVED - 104h	104h
FAULT_MSG_INVALID_PARAMETER_VALUE - 105h	105h
FAULT_MSG_POWER_TOO_LOW - 106h	106h
FAULT_UNIMPLEMENTED_FEATURE - 109h	109h
FAULT_INVALID_BAUD_RATE - 10Ah	10Ah
FAULT_INVALID_REGION – 10Bh	10Bh

FAULT_MSG_WRONG_NUMBER_OF_DATA – 100h

Cause

If the data length in any of the Host-to-M5e/M5e-Compact messages is less than or more than the number of arguments in the message, the reader returns this message.

Solution

Make sure the number of arguments matches the data length.

FAULT_INVALID_OPCODE – 101h

Cause

The opCode received is invalid or not supported in the currently running program (bootloader or main application) or is not supported in the current version of code.

Solution

Check the following:

- Make sure the command is supported in the currently running program.
- Check the documentation for the opCode the host sent and make sure it is correct and supported.
- Check the previous module responses for an assert (0x7F0X) which will reset the module into the bootloader.

FAULT_UNIMPLEMENTED_OPCODE – 102h

Cause

Some of the reserved commands might return this error code.

This does not mean that they always will do this since ThingMagic reserves the right to modify those commands at anytime.

Solution

Check the documentation for the opCode the host sent to the reader and make sure it is supported.

FAULT_MSG_POWER_TOO_HIGH – 103h

Cause

A message was sent to set the read or write power to a level that is higher than the current HW supports.

Solution

Check the HW specifications for the supported powers and insure that the level is not exceeded.

The M5e 1 Watt units support power from 5 dBm to 30 dBm.

The M5e-Compact units support power from 10 dBm to 23 dBm.

FAULT_MSG_INVALID_FREQ_RECEIVED - 104h

Cause

A message was received by the reader to set the frequency outside the supported range

Solution

Make sure the host does not set the frequency outside this range or any other locally supported ranges.

FAULT_MSG_INVALID_PARAMETER_VALUE - 105h

Cause

The reader received a valid command with an unsupported or invalid value within this command.

For example, currently the module supports two antennas, 1 and 2. If the module receives a message with an antenna value other than 1 or 2, it returns this error.

Solution

Make sure the host sets all the values in a command according to the values published in this document.

FAULT_MSG_POWER_TOO_LOW - 106h

Cause

A message was received to set the read or write power to a level that is lower than the current HW supports.

Solution

Check the HW specifications for the supported powers and insure that level is not exceeded. The M5e supports powers between 5 and 30 dBm. The M5e-Compact units support power from 10 dBm to 23 dBm.

FAULT_UNIMPLEMENTED_FEATURE - 109h

Cause

Attempting to invoke a command not supported on this firmware or hardware.

Solution

Check the command being invoked against the documentation.

FAULT_INVALID_BAUD_RATE - 10Ah

Cause

When a **Set Baud Rate** (0x06h) command is issued for a rate that is not specified in the Baud Rate table, this error message is returned.

Solution

Check the table of specific baud rates and select a baud rate. Send the baud rate in the hex format. See Set Baud Rate (06h).

FAULT_INVALID_REGION – 10Bh

Cause

The region code specified in Set Current Region (97h) is invalid or not supported in the currently running application program or not supported in the current version of code.

Solution

Check the supported Region codes for the hardware in use.

Bootloader Faults

The following table lists the common faults discussed in this section.

Fault Message	Code
FAULT_BL_INVALID_IMAGE_CRC	200h
FAULT_BL_INVALID_APP_END_ADDR	201h

FAULT_BL_INVALID_IMAGE_CRC – 200h

Cause

When a **Verify Image CRC** (0x08), or **Boot Firmware** (0x02) command is issued, the reader checks the image stored in flash and returns this error if the calculated CRC is different than the one stored in flash.

Solution

The exact reason for the corruption could be that the image loaded in flash was corrupted during the transfer or corrupted for some other reason.

To fix this problem, reload the application code in flash.

FAULT_BL_INVALID_APP_END_ADDR – 201h

Cause

When a **Verify Image CRC** (0x08), or **Boot Firmware** (0x02) command is issued, the reader checks the image stored in flash and returns this error if the last word stored in flash does not have the correct address value.

Solution

The exact reason for the corruption could be that the image loaded in flash got corrupted during the transfer or, corrupted for some other reason.

To fix this problem, reload the application code in flash.

Flash Faults

The following table lists the common faults discussed in this section.

Fault Message	Code
FAULT_FLASH_BAD_ERASE_PASSWORD – 300h	300h
FAULT_FLASH_BAD_WRITE_PASSWORD – 301h	301h
FAULT_FLASH_UNDEFINED_ERROR – 302h	302h
FAULT_FLASH_ILLEGAL_SECTOR – 303h	303h
FAULT_FLASH_WRITE_TO_NON_ERASED_AREA – 304h	304h
FAULT_FLASH_WRITE_TO_ILLEGAL_SECTOR – 305h	305h
FAULT_FLASH_VERIFY_FAILED – 306h	306h

FAULT_FLASH_BAD_ERASE_PASSWORD – 300h

Cause

A command was received to erase some part of the flash but the password supplied with the command was incorrect.

Solution

Make sure that you have the correct password for the flash sector.

Go to the Accessing the Flash section for more information about the flash passwords and sectors.

FAULT_FLASH_BAD_WRITE_PASSWORD – 301h

Cause

A command was received to write some part of the flash but the password supplied with the command was not correct.

Solution

Make sure that you have the correct password for the flash sector.

Check the Accessing the Flash for more information about the flash passwords and sectors.

FAULT_FLASH_UNDEFINED_ERROR – 302h

Cause

This is an internal error and it is caused by a software problem in module.

Solution

Contact support at support@thingmagic.com.

FAULT_FLASH_ILLEGAL_SECTOR – 303h

Cause

An erase or write flash command was received with the sector value and password not matching.

Solution

Make sure that you have the correct password for the flash sector.

Go to Accessing the Flash for more information about the flash passwords and sectors.

FAULT_FLASH_WRITE_TO_NON_ERASED_AREA – 304h

Cause

The module received a write flash command to an area of flash that was not previously erased.

Solution

Erase that sector of flash and then, try and rewrite to it.

FAULT_FLASH_WRITE_TO_ILLEGAL_SECTOR – 305h

Cause

The module received a write flash command to write across a sector boundary that is prohibited.

Solution

If the data spans two sectors, separate the data into two messages.

FAULT_FLASH_VERIFY_FAILED – 306h

Cause

The module received a write flash command that was unsuccessful because data being written to flash contained an uneven number of bytes.

Solution

Verify that the data being sent is an even number of bytes.

Protocol Faults

The following table lists the common faults discussed in this section.

Fault Message	Code
FAULT_NO_TAGS_FOUND – 400h	400h
FAULT_NO_PROTOCOL_DEFINED – 401h	401h
FAULT_INVALID_PROTOCOL_SPECIFIED – 402h	402h
FAULT_WRITE_PASSED_LOCK_FAILED – 403h	403h
FAULT_PROTOCOL_NO_DATA_READ – 404h	404h
FAULT_AFE_NOT_ON – 405h	405h
FAULT_PROTOCOL_WRITE_FAILED – 406h	406h
FAULT_NOT_IMPLEMENTED_FOR_THIS_PROTOCOL – 407h	407h
FAULT_PROTOCOL_INVALID_WRITE_DATA – 408h	408h
FAULT_PROTOCOL_INVALID_ADDRESS – 409h	409h
FAULT_GENERAL_TAG_ERROR – 40Ah	40Ah
FAULT_DATA_TOO_LARGE – 40Bh	40Bh
FAULT_PROTOCOL_INVALID_KILL_PASSWORD – 40Ch	40Ch
FAULT_PROTOCOL_KILL_FAILED - 40Eh	40Eh
FAULT_PROTOCOL_BIT_DECODING_FAILED - 40Fh	40Fh
FAULT_PROTOCOL_INVALID_EPC – 410h	410h
FAULT_PROTOCOL_INVALID_NUM_DATA – 411h	411h
FAULT_GEN2 PROTOCOL_OTHER_ERROR - 420h	420h
FAULT_GEN2 PROTOCOL_MEMORY_OVERRUN_BAD_PC - 423h	423h
FAULT_GEN2 PROTOCOL_MEMORY_LOCKED - 424h	424h
FAULT_GEN2 PROTOCOL_INSUFFICIENT_POWER - 42Bh	42Bh
FAULT_GEN2 PROTOCOL_NON_SPECIFIC_ERROR - 42Fh	42Fh
FAULT_GEN2 PROTOCOL_UNKNOWN_ERROR - 430h	430h

FAULT_NO_TAGS_FOUND – 400h

Cause

A command was received (such as like read, write, or lock) but the operation failed. There are many reasons that can cause this error to occur.

Here is a list of possible reasons that could be causing this error:

- No tag in the RF field
- Read/write power too low
- Antenna not connected
- Tag is weak or dead

Solution

Make sure there is a good tag in the field and all parameters are set up correctly. The best way to check this is to try few tags of the same type to rule out a weak tag. If none passed, then it could be SW configuration such as protocol value, antenna, and so forth, or a placement configuration like a tag location.

FAULT_NO_PROTOCOL_DEFINED – 401h

Cause

A command was received to perform a protocol command but no protocol was initially set. The reader powers up with no protocols set.

Solution

A **Set Current Tag Protocol** (63h) command must be sent followed by resending the desired command.

FAULT_INVALID_PROTOCOL_SPECIFIED – 402h

Cause

A **Set Current Tag Protocol** (63h) command was received for a protocol value that is not supported with the current version of SW.

Solution

This value is invalid or this version of SW does not support the protocol value. Check the documentation for the correct values for the protocols in use.

FAULT_WRITE_PASSED_LOCK_FAILED – 403h

Cause

During a Write Tag Data for ISO18000-6B or UCODE, if the lock fails, this error is returned. The write command passed but the lock did not. This could be a bad tag.

Solution

Try to write a few other tags and make sure that they are placed in the RF field.

FAULT_PROTOCOL_NO_DATA_READ – 404h

Cause

A **Read Tag ID** or **Data** command was sent but did not succeed.

Solution

The tag used has failed or does not have the correct CRC. Try to read a few others to check the HW/SW configuration.

FAULT_AFE_NOT_ON – 405h

Cause

A command was received for an operation, like read or write, but the RF cannot turn on because the Region and/or Protocol have not been set..

Solution

Call Set Current Region (97h) and Set Current Tag Protocol (93h) to set the region of operation and tag protocol, respectively.

FAULT_PROTOCOL_WRITE_FAILED – 406h

Cause

This fault can occur when an operation such as write, lock, kill, set password, or initialize, fails. There are many reasons for failure.

Solution

Check that the tag is good and try another operation on a few more tags.

FAULT_NOT_IMPLEMENTED_FOR_THIS_PROTOCOL – 407h

Cause

A command was received which is not supported by a protocol.

Solution

Check the documentation for the supported commands and protocols.

FAULT_PROTOCOL_INVALID_WRITE_DATA – 408h

Cause

In EPC0+, the first two bits determine the tag ID length. If the first two bits are 0b00, then the tag ID must be 96-bits. Otherwise the tag ID is 64 bits.

Solution

Make sure that the first two bit have the correct values depending in the Tag ID length.

FAULT_PROTOCOL_INVALID_ADDRESS – 409h

Cause

A command was received attempting to access an invalid address in the tag data address space.

Solution

Make sure that the address specified is within the scope of the tag data address space and available for the specific operation. The protocol specifications contain information about the supported addresses.

FAULT_GENERAL_TAG_ERROR – 40Ah

Cause

This error is used by the M5e GEN2 module. This fault can occur if the read, write, lock, or kill command fails. This error can be internal or functional.

Solution

Make a note of the operations you were performing and contact ThingMagic at http://support.thingmagic.com

FAULT_DATA_TOO_LARGE – 40Bh

Cause

A command was received to Read Tag Data with a data value larger than expected or it is not the correct size.

Solution

Check the size of the data value in the message sent to the reader.

FAULT_PROTOCOL_INVALID_KILL_PASSWORD – 40Ch

Cause

An incorrect kill password was received as part of the **Kill Tag** (26h) command.

Solution

Check the password.

FAULT_PROTOCOL_KILL_FAILED - 40Eh

Cause

Attempt to kill a tag failed for an unknown reason

Solution

Check tag is in RF field and the kill password.

FAULT_PROTOCOL_BIT_DECODING_FAILED - 40Fh

Cause

Attempt to operate on a tag with an EPC length greater than the Maximum EPC length setting.

Solution

Call Set Reader Configuration(9Ah) to set the Max EPC to 496 bits..

FAULT_PROTOCOL_INVALID_EPC – 410h

Cause

This error is used by the M5e GEN2 module indicating an invalid EPC value has been specified for an operation. This fault can occur if the read, write, lock, or kill command fails.

Solution

Check the EPC value that is being passed in the command resulting in this error.

FAULT_PROTOCOL_INVALID_NUM_DATA – 411h

Cause

This error is used by the M5e GEN2 module indicating invalid data has been specified for an operation. This fault can occur if the read, write, lock, or kill command fails.

Solution

Check the data that is being passed in the command resulting in this error.

FAULT_GEN2 PROTOCOL_OTHER_ERROR - 420h

FAULT_GEN2_PROTOCOL_MEMORY_OVERRUN_BAD_PC - 423h

FAULT_GEN2 PROTOCOL_MEMORY_LOCKED - 424h

FAULT_GEN2 PROTOCOL_INSUFFICIENT_POWER - 42Bh

FAULT_GEN2 PROTOCOL_NON_SPECIFIC_ERROR - 42Fh

FAULT_GEN2 PROTOCOL_UNKNOWN_ERROR - 430h

Analog Hardware Abstraction Layer Faults

FAULT_AHAL_INVALID_FREQ – 500h

Cause

A command was received to set a frequency outside the specified range.

For example, in North America the frequency range is from 902 MHz to 928 MHz.

Solution

Check the values you are trying to set and be sure that they fall within this range.

FAULT_AHAL_CHANNEL_OCCUPIED – 501h

Cause

With LBT enabled an attempt was made to set the frequency to an occupied channel.

Solution

Try a different channel.

FAULT_AHAL_TRANSMITTER_ON – 502h

Cause

Checking antenna status while CW is on is not allowed.

Solution

Do not perform antenna checking when CW is turned on.

FAULT_ANTENNA_NOT_CONNECTED – 503h

Cause

An attempt was made to transmit on an antenna which did not pass the antenna detection when antenna detection was turned on.

Solution

Connect a detectable antenna (antenna must have some DC resistance).

FAULT_TEMPERATURE_EXCEED_LIMITS – 504h

Cause

The module has exceeded the maximum or minimum operating temperature and will not allow an RF operation until it is back in range.

Solution

Take steps to resolve thermal issues with module:

- Reduce duty cycle
- Add heat sink
- Use Low Power Mode

FAULT_HIGH_RETURN_LOSS – 505h

Cause

The module has detected a high return loss and has ended RF operation to avoid module damage.

Solution

Take steps to resolve high return loss on receiver:

- Make sure antenna VSWR is within module specifications
- Make sure antennas are correctly attached before transmitting

- Check environment to ensure no occurrances of high signal reflection back at antennas.

FAULT_AHAL_INVALID_ANTENA_CONFIG – 507h

Cause

An attempt to set an antenna configuration that is not valid based on the current *GPIO as Antenna Switch* setting applied using Set Reader Configuration(9Ah).

Solution

Use the correct antenna setting or change the reader configuration.

Tag ID Buffer Faults

The following table lists the common faults discussed in this section.

Fault Message	Code
FAULT_TAG_ID_BUFFER_NOT_ENOUGH_TAGS_AVAILABLE – 600h	600h
FAULT_TAG_ID_BUFFER_FULL – 601h	601h
FAULT_TAG_ID_BUFFER_REPEATED_TAG_ID – 602h	602h
FAULT_TAG_ID_BUFFER_NUM_TAG_TOO_LARGE – 603h	603h

FAULT_TAG_ID_BUFFER_NOT_ENOUGH_TAGS_AVAILABLE – 600h

Cause

A command was received to get a certain number of tag ids from the tag id buffer. The reader contains less tag ids stored in its tag id buffer than the number the host is sending.

Solution

Send a **Get Tag ID Buffer** command to ascertain how many tags are in the buffer. You can get the exact number of tags as long as they are less than or equal to the number returned by the previous command.

FAULT_TAG_ID_BUFFER_FULL – 601h

Cause

The tag id buffer is full.

Solution

Send **Clear Tag ID Buffer** or **Get Tag ID Buffer** command with the number of tags to read, to get more space See Get Tag Buffer (29h) for more information.

FAULT_TAG_ID_BUFFER_REPEATED_TAG_ID – 602h

Cause

The module has an internal error. One of the protocols is trying to add an existing TagID to the buffer.

Solution

Report this problem to ThingMagic at http://support.thingmagic.com.

FAULT_TAG_ID_BUFFER_NUM_TAG_TOO_LARGE – 603h

Cause

The module received a request to retrieve more tags than is supported by the current version of the software.

Solution

Locate the maximum number of supported tags in the TagID buffer in this document Get Tag Buffer (29h).

System Errors

FAULT_SYSTEM_UNKNOWN_ERROR – 7F00h

Cause

The error is internal.

Solution

Make note of the operations you were executing and contact ThingMagic at http://support.thingmagic.com.

FAULT_TM_ASSERT_FAILED – 7F01h

Cause

An unexpected Internal Error has occurred.

Solution

The error will cause the module to switch back to Bootloader mode. When this occurs make note of the operations you were executing, save FULL error response and contact ThingMagic at support@thingmagic.com.

Appendix D: Deprecated and Modified Commands

The following contains the deprecated versions of Serial commands grouped by the release they were deprecated in. These commands are still supported for backward compatibility. For new development the new formats for each, as specified in Command Set, should be used.

Release Version 1.0.34

Read Tag Single (21h)

If a tag is read using the currently set tag protocol, its EPC is returned in the tag ID section of the packet. If no tag is read, a fault code is reported. The **Read Tag Single** command takes a 16-bit timeout value in milliseconds.

The following example shows a timeout of 1000 ms (0x03E8):

FF	02	21	03 E8	D5 09
SOH	Length	OpCode	Timeout (ms)	CRC

The response to this command is slightly different depending upon the number of bits in the tag ID of the current protocol that is selected. The general response format is shown here:

FF	M+2	21	00 00	M bytes	?? ??	?? ??
SOH	Length	OpCode	Status	Tag ID	TagCRC	CRC

For example, the following response packet for an EPC0 64-bit tag is sent back. Notice that the Tag CRC follows the tag ID field for all protocols:

FF	0A	21	00 00	C8 05 07 A8 00 84 C4 FF	9E E0	F7 25
SOH	Length	OpCode	Status	Tag ID	[1]Tag CRC	[2]CRC

1.Tag CRC – Calculated on the tag ID field only.
2.CRC – Calculated on the entire message packet.

Note

It is important to have a valid protocol selected. To select a protocol, use the **Set Tag Protocol** command. This command will not work correctly if there are multiple tags in the RF field, use the **Read Tag Multiple** command instead.

Read Tag ID Multiple (22h)

The **Read Tag ID Multiple** command performs a search for the specified period of time then returns the number of tags that have been found. Afterwards, multiple **Get Tag ID Buffer** commands can be sent to receive the found tag IDs. Each protocol defines its own search method.

FF	02	22	03 E8	E5 6A
SOH	Length	OpCode	Timeout (ms)	CRC

The return provides the following:

FF	01	22	00 00	02	46 BA
SOH	Length	OpCode	Status	# Tag IDs Found	CRC

Write Tag Data (24h)

The **Write Tag Data** command writes into the data section of the tag. To write the tag ID, use the **Write Tag ID** command.

FF	M + N + 7	24	03 E8	00	00 00 00 00	M bytes	N bytes	?? ??
SOH	Length	OpCode	Timeout (ms)	Lock Bit	Address	Tag ID (M bytes)	Data to Write	CRC

GEN2

For Gen2, if Option=00, the write tag data is performed without the select option with the following format:

FF	N+8	24	03 E8	00	00 00 00 02	01	N bytes	?? ??
SOH	Length	OpCode	Timeout (ms)	Option	Address	MemBank	Data to Write	CRC

In the previous example, the reader can singulate a tag with Q=0 and would try to write N bytes of data into the first tag it can see, starting at the address 0x2 of EPC MemBank (the first word of the tag's EPC).

If Option=01, the write tag data is performed with the select option that enables you to choose a tag from a population of tags based on its Tag ID with the following format:

FF	M+N+13	24	03 E8	01	00 00 00 02	00	00 00 00 00	
SOH	Length	OpCode	Timeout (ms)	Option	Address	MemBank	Access Password	

60h	M bytes	N bytes	??	??
Tag ID Length	Tag ID (M bytes)	Data to Write	CRC	

In this example, if N=4, the reader issues a **Select** command to singulate the desired tag and then would write 4 bytes of data into the tag memory of a 96-bit tag with the specified Tag ID, starting at the address 0x2 of the Reserved MemBank (the first word of the Kill Password of the tag).

Note

In both examples, N is a variable value that is an even, nonzero number less than 32.

The 32-bit Address field is the offset from the MemBank origin, in 16-bit words, where the contents of the Data field is written. It corresponds to the *WordPtr* argument in the Gen2 command specification.

The 8-bit MemBank field specifies which of the tag's memory banks is to be addressed. It corresponds to the *Gen2* argument of the same name. Its values are as follows.

MemBank	Description
0x00	Reserved
0x01	EPC
0x02	TID
0x03	User

For more information about the MemBank structure and the memory map of the Gen2 tag, please refer to the EPC Global GEN2 protocol specification. This command is not supported by the EPC1 protocol.

Lock Tag (25h)

The **Lock Tag** command locks a specific address in the tag data section. The ID of the tag, the address to lock, and a password for killing the tag need to be provided.

FF	M + P + 6	25	03 E8	P bytes	00 00 00 08	M bytes	?? ??
SOH	Length	OpCode	Timeout (ms)	Password	Address	Tag ID (M bytes)	CRC

For the GEN2 protocol, the command format is as follows.

FF	0A	25	03 E8	11 22 33 44	11 22	11 22	?? ??
SOH	Length	OpCode	Timeout (ms)	Access Password	Mask Bits	Action Bits	CRC

The **Lock** command sends a command sequence including **Lock** described in the Gen2 specification. The 32-bit Access Password argument must contain the access password that was written previously to word addresses 0x02 and 0x03 (bit addresses 0x20-0x3F) of the tag.

The Mask and Action bits correspond to the identically named fields described in Section 6.3.2.10.3.5 of the Gen2 specification. Each of these 10-bit fields is right-aligned (LSB-aligned) in the corresponding 16-bit argument to the serial command, as shown in the following table. These fields are passed directly to the tag.

	First Byte (0x11 above)								Second Byte (0x22 above)							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Unused						Kill Pwd		Access Pwd		EPC Mem		TID Mem		User Mem	
Mask	X	X	X	X	X	X	L	L	L	L	L	L	L	L	L	L
Action	X	X	X	X	X	X	R/W	Perm	R/W	Perm	W	Perm	W	Perm	W	Perm

Where a Mask bit is specified as “L,” a “1” in that bit indicates that the corresponding bit in the **Action** field is set or cleared. A “0” indicates that the corresponding bit in the **Action** field should be ignored.

Where an Action bit is specified as “R/W,” a “1” indicates that reading and writing should be locked. A “W” indicates that only writing should be locked.

Note

Adding ‘0000’ in the **Action** field unlocks the memory bank.

A “Perm” bit sets the permalock state of the corresponding “R/W” or “W” bit, which permanently programs the lock status of that bit on the tag. An Action bit has no effect unless the corresponding Mask bit is set. An “X” in the above table indicates an ignored bit. For more detailed information about these fields, please refer to the Gen2 specification.

A tag whose access password is zero transitions directly from the acknowledged state to the secured state without first stopping in the open state. No password check is required. In other words, locking a tag has no effect if its access password is set to zero (unless it is permalocked). The access password can be set using the **Write Tag Data** command.

Kill Tag (26h)

The **Kill Tag** command kills a tag. Since each protocol has a different format for the kill command, this is highly protocol dependent. As each new kill command becomes available, the message structure will be documented in this section.

For the GEN2 protocol, the following command format is used.

FF	07	26	03 E8	11 22 33 44	00	?? ??
SOH	Length	OpCode	Timeout (ms)	Kill Password	RFU	CRC

The kill password must correspond to the kill password programmed in the tag.

Note

If the tag's kill password is set to 0, the protocol does not allow the tag to be killed. A non-zero kill password must be set (using the **Write Tag Data** command) before the kill command succeeds. The RFU field should be set to 0.

Read Tag Data (28h)

The **Read Tag Data** command reads data from the tag. Because each protocol has different formats for reading / writing data, the address field may have different meanings.

The general format for the **Read Tag Data** command is shown in the following example. The tag ID field can be either 8 or 12 bytes, depending on whether it is a 64-bit or 96-bit tag.

FF	M + 6	28	03 E8	00 00 00 10	M bytes	?? ??
SOH	Length	OpCode	Timeout (ms)	Address	Tag ID (M bytes)	CRC

GEN2 Command and Response

The **Read Tag Data** command for the GEN2 protocol has the following format. This example reads the first 16-bit word of the EPC stored in a tag using the GEN2 protocol **Read** command.

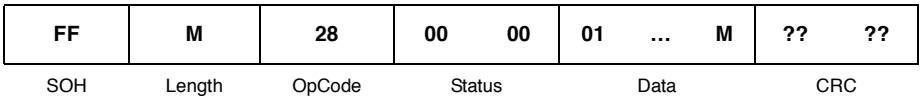
FF	08	28	03 E8	01	00 00 00 02	01	?? ??
SOH	Length	OpCode	Timeout (ms)	MemBank	Address	WordCount	CRC

Read Tag Data Command

The **MemBank**, **Address**, and **WordCount** fields correspond to the *MemBank*, *WordPtr*, and *WordCount* arguments to the GEN2 protocol **Read** command, respectively. The MemBank options are listed in the Write Tag Data (24h) section of this document.

Refer to the EPC Global GEN2 protocol specification for more details about the tag memory map.

The response to the **Read Data** command for the GEN2 protocol has the following format.

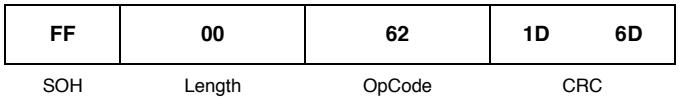


Read Tag Data Response

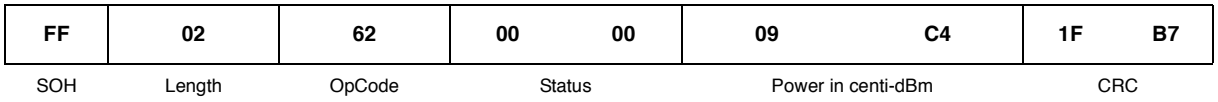
The length (M) of the data returned is twice the WordCount value in the original command because the return packet length is counted in bytes.

Get Read TX Power (62h)

The **Get Read TX Power** command returns the current TX power for reading tags, in centi-dBm:



Microprocessor reply is similar to the **Set Read TX Power** command. In this example, 0x09C4 corresponds to 2500, which is 25.00 dBm. The default read power is set to 30 dBm:



Get Write TX Power (64h)

The **Get Write TX Power** command gets the current power level used for TX write commands:

FF	00	64	1D 6B
SOH	Length	OpCode	CRC

Microprocessor reply is similar to **Set Write TX Power**. The default Write TX power is set to 30 dBm:

FF	02	64	00 00	0A 5A	AE 89
SOH	Length	OpCode	Status	Power in centi-dBm	CRC

Get Current Region (67h)

The **Get Current Region** command gets the current region set in the reader. Currently available region codes are shown in the following table. Details on the Regulatory Compliance for each region can be found in Regional Support::

Region codes

Region	LBT Support/ Default	Code
NA	No	0x01
EU	Yes/On	0x02
EU2	No	0X07
EU3	Yes/Off	0x08
KR	No	0x03
Open	Yes/Off	0xFF

To get the current region, send the following command:

FF	00	67	1D 68
SOH	Length	OpCode	CRC

The command generates the following reply, which indicates that the reader is set to the NA region:

FF	01	67	00 00	01	B4 80
SOH	Length	OpCode	Status	Region	CRC

Get Transmit Mode (6Ah)

The **Get Transmit Mode** command returns the tranmit modethe M5e is current set to. The available transmit modes are shown in the following table:

Available Transmit Modes

Hex	Transmit Mode
0x0000	High performance mode (disables low power mode)
0x0001	Low power mode (disables high performance mode)

Send the following command to get the currrent transmit mode:

FF	00	6A	1D 65
SOH	Length	OpCode	CRC

The example reply shows the current transmit mode setting:

FF	02	6A	00 00	00 01	BF D3
SOH	Length	OpCode	Status	Transmit Mode	CRC

Set Current Region (97h)

The **Set Current Region** command sets the current region for use in the reader. The list of region codes are found in Region codes. However, the module is able to support only a limited subset of those regions.

Setting the region performs the following:

1. Frequency hop table is set to the default for the region.
2. Power limits are applied to ensure unit does not send out too much power.
3. Any other region specific settings are enforced.

For example, to set the region to KR:

FF	01	97	03	4B	BE
SOH	Length	OpCode	Region	CRC	

Set Transmit Mode (9Ah) [M5e Only]

Use the **Set Transmit Mode** command to set the transmit mode the M5e will use. The available transmit modes are shown in the following table:

Available Transmit Modes

Hex	Transmit Mode
0x0000	High performance mode (disables low power mode)
0x0001	Low power mode (disables high performance mode)

Send the following to enable the Low power transmit mode:

FF	02	9A	00	01	C0	50
SOH	Length	OpCode	Transmit Mode		CRC	

Note

The default setting is high performance mode (disables low power mode).

Release Version 1.0.37.27

Set Reader Configuration(9Ah)

For the deprecated version of this command see Appendix D: Set Transmit Mode (9Ah) [M5e Only]

The **Set Reader Configuration** command is used to set several configuration options on the reader. The configuration options, defined in the table below are enabled bitwise, although the option value is specified in hexidecimal (2 bytes). One example configuration and the corresponding Options value is specified in the command example below. For other combinations the hex value for Options will need to be calculated by OR'ing the bits for each setting.

Available Configuration Options

Option Bit/ Mode	Bit Value (4 LSBs shown)	Reader Configuration setting
Bit 0 Transmit Mode	$XXX0_2$ (Default)	High Performance Mode (disables low power mode)
	$XXX1_2$	Low Power Mode (disables high performance mode)
Bit 1 EPC Length	$XX0X_2$ (Default)	Maximum EPC length of 96 bits
	$XX1X_2$	Maximum EPC length of 496 bits

Send the following to enable the Low power transmit mode and support for EPC values up to 496 bits:

FF	02	9A	00	03	C0	52
SOH	Length	OpCode	Options		CRC	

Get Reader Configuration(6Ah)

For the deprecated version of this command see Appendix D: Get Transmit Mode (6Ah)

The **Get Reader Configuration** command returns the current reader configuration as defined in Set Transmit Mode (9Ah) [M5e Only]:

Send the following command to get the current transmit mode:

FF	00	6A	1D 65
SOH	Length	OpCode	CRC

The example reply shows the current reader configuration Options setting:

FF	02	6A	00 00	00 01	BF D3
SOH	Length	OpCode	Status	Options	CRC

Appendix E: Environmental Considerations

This Appendix details environmental factors that should be considered relating to reader performance and survivability.

ElectroStatic Discharge (ESD) Considerations

W A R N I N G !

The M5e antenna ports may be susceptible to damage from Electrostatic Discharge (ESD). Equipment failure can result if the antenna or communication ports are subjected to ESD. Standard ESD precautions should be taken during installation to avoid static discharge when handling or making connections to the M6 reader antenna or communication ports. Environmental analysis should also be performed to ensure static is not building up on and around the antennas, possibly causing discharges during operation.

ESD Damage Overview

In M5e-based reader installations where readers have failed without known cause, based on anecdotal information ESD has been found to be the most common cause. Failures due to ESD tend to be in the M5E power amplifier section (PA). PA failures typically manifest themselves at the software interface in the following ways:

- RF operations (read, write, etc.) respond with **status code 0x7F01** - indicating a a fatal error. This is typically due the the module not being able to reach the target power level due to PA damage.
- RF operations (read, write, etc.) respond with **status code 0x0505** even when a known good antenna is attached.
- Unexpected **status code 0x0101**, indicating command not supported, when that command had worked just fine shortly before. The reason a command becomes suddenly not supported is that the reader, in the course of its self protection routines,

has returned to the bootloader to prevent any further damage. This jump to boot loader caused by power amp damage occurs at the start of any read tag commands.

Ultimately determining that ESD is the root cause of failures is difficult because it relies on negative result experiments, i.e. it is the lack of failure after a configuration change, rather than a positive flag wave that says “I’m ESD”. Such flag waves are sometimes, but only sometimes, available at the unpackaged transistor level under high power microscopy. The remoteness of microscopic examination from the installed field failures is indicative of the high cost of using such analysis methods for chasing down ESD issues. Therefore most ESD issue resolutions will be using the negative result experiments to determine success.

ESD discharges come with a range of values, and like many things in life there is the “matter of degree”. For many installations, the M5E has been successfully deployed and operates happily. For these, there is no failure issue, ESD or otherwise. For a different installation that with bare M5E, has a failure problem from ESD, there will be some distribution of ESD intensities occurring. Without knowledge of a limit in the statistics of those intensities, there may always be the bigger zap waiting in the wings. For the bare M5E equipped with the mitigation methods described below, there will always be the rouge ESD discharge that exceeds any given mitigation, and results in failure. Fortunately, many installations will have some upper bound on the value of ESD events given the geometry of that installation.

Several sequential steps are recommended for a) determining the ESD is the likely cause of a given group of failures, and b) enhancing the M5E’s environment to eliminate ESD failures. The steps vary depending on the required M5E output power in any given application.

Identifying ESD as the Cause of Damaged Readers

The following are some suggested methods to determine if ESD is a cause of reader failures, i.e. ESD diagnostics. Please remember- some of these suggestions have the negative result experiment problem.

- Return failed units for analysis. Analysis should be able to say if it is the power amplifier that has in fact failed, but won’t be able to definitively identify that the cause is ESD. However, ESD is one of the more common causes of PA failure.
- Measure ambient static levels with static meter. *AlphaLabs SVM2* is such a meter, but there are others. You may be surprised at the static potentials floating detected. However, high static doesn’t necessarily mean discharges, but should be considered cause for further investigation. High levels that keep changing are highly indicative of discharges.
- Touch some things around the antenna, and operating area. If you feel static discharges, that qualitatively says quite a bit about what is in front of the antenna.

What actually gets to the M5E is also strongly influenced by the antenna installation, cabling, and grounding discussed above.

- Use the mean operating time statistic before and after one or more of the changes listed below to quantitatively determine if the change has resulted in an improvement. Be sure to restart your statistics after the change.

Common Installation Best Practices

The following are common installation best practices which will ensure the readers isn't being unnecessarily exposed to ESD in even low risk environments. These should be applied to all installations, full power or partial power, ESD or not:

- Insure that M5E, M5E enclosing housing (e.g. Vega reader housing), and antenna ground connection are all grounded to a common low impedance ground.
- Verify R-TNC knurled threaded nuts are tight and stay tight. Don't use a thread locking compound that would compromise the grounding connection of the thread to thread mate. If there is any indication that field vibration might cause the R-TNC to loosen, apply RTV or other adhesive externally.
- Use antenna cables with double shield outer conductors, or even full metallic shield semirigid cables. ThingMagic specified cables are double shielded and adequate for most applications. ESD discharge currents flowing ostensibly on the outer surface of a single shield coaxial cable have been seen to couple to the inside of coaxial cables, causing ESD failure. Avoid RG-58. Prefer RG-223.
- Minimize ground loops in coaxial cable runs to antennas. Having the M5E and antenna both tied to ground (per item 1) leads to the possibility of ground currents flowing along antenna cables. The tendency of these currents to flow is related to the area of the conceptual surface marked out by the antenna cable and the nearest continuous ground surface. When this conceptual surface has minimum area, these ground loop current are minimized. Routing antenna cables against grounded metallic chassis parts helps minimize ground loop currents.
- Keep the antenna radome in place. It provides significant ESD protection for the metallic parts of the antenna, and protects the antenna from performance changes due to environmental accumulation.
- Keep careful track of serial numbers, operating life times, numbers of units operating. You need this information to know that your mean operating life time is. Only with this number will you be able to know if you have a failure problem in the first place, ESD or otherwise. And then after any given change, whether things have improvement or not. Or if the failures are confined to one instantiation, or distributed across your population.

Raising the ESD Threshold

For applications where full M5E power is needed for maximum tag read range and ESD is suspected the following components are recommended additions to the installation to raise the level of ESD the reader can tolerate:

- Select or change to an antenna with all radiating elements grounded for DC. The MTI MT-262031-T(L,R)H-A is such an antenna. The Laird IF900-SF00 and CAF95956 are not such antennas. The grounding of the antenna elements dissipates static charge leakage, and provides a high pass characteristic that attenuates discharge events. (This also makes the antenna compatible with the M5e antenna detect methods.)
- Install a Minicircuits SHP600+ high pass filter in the cable run at the M5E (or Vega or other finished reader) end. This additional component will reduce transmit power by 0.4 dB which may affect read range in some critical applications. However the filter will significantly attenuate discharges and improve the M5E ESD survival level.

  Note

  The SHP600+ is not rated for the full +30 dBm output of the M5E of Vega reader at +85 degree C. Operation at reduced temperature has been anecdotally observed to be OK, but has not been fully qualified by ThingMagic.

- Install a Diode Clamp* circuit immediately outboard from the SHP600 filter. This will reduce transmit power by an additional 0.4 dB, but in combination with the SHP600 will further improve the M5E ESD survival level. * Not yet productized. Needs DC power, contact support@thingmagic.com for details.

Further ESD Protection for Reduced RF Power Applications

In addition to the protective measures recommended above, for applications where reduced M5E RF power is acceptable and ESD is suspected the following protective measures can also be applied:

- Install a one watt attenuator with a decibel value of +30 dBm minus the dBm value needed for tag power up. Then run the reader at +30 dBm instead of reduced transmit power. This will attenuate inbound ESD pulses by the installed decibel value, while keeping the tag operation generally unchanged. Attenuators of 6 dB have been shown to not adversely effect read sensitivity. Position the attenuator as close to the M5E as feasible.
- As described above add the SHP600 filter immediately adjacent to the attenuator, on the antenna side.
- Add Diode Clamp, if required, adjacent to the SHP600, on the antenna side.

Variables Affecting Performance

Reader performance may be affected by the following variables, depending on the site where your Reader is being deployed:

Environmental

Reader performance may be affected by the following environmental conditions:

- Metal surfaces such as desks, filing cabinets, bookshelves, and wastebaskets may enhance or degrade Reader performance.
- Antennas should be mounted far away from metal surfaces that may adversely affect the system performance.
- Devices that operate at 900 MHz, such as cordless phones and wireless LANs, can degrade Reader performance. The Reader may also adversely affect the performance of these 900 MHz devices.
- Moving machinery can interfere the Reader performance. Test Reader performance with moving machinery turned off.
- Fluorescent lighting fixtures are a source of strong electromagnetic interference and if possible should be replaced. If fluorescent lights cannot be replaced, then keep the Reader cables and antennas away from them.
- Coaxial cables leading from the Reader to antennas can be a strong source of electromagnetic radiation. These cables should be laid flat and not coiled up.

Tag Considerations

There are several variables associated with tags that can affect Reader performance:

- Application Surface: Some materials, including metal and moisture, interfere with tag performance. Tags applied to items made from or containing these materials may not perform as expected.

- Tag Orientation: Reader performance is affected by the orientation of the tag in the antenna field. The ThingMagic antenna is circularly polarized, so it reads face-to but not edge-to.
- Tag Model: Many tag models are available. Each model has its own performance characteristics.

Multiple Readers

The Reader adversely affect performance of 900 MHz devices. These devices also may degrade performance of the Reader.

- Antennas on other Readers operating in close proximity may interfere with one another, thus degrading performance of the Readers.
- Interference from other antennas may be eliminated or reduced by using either one or both of the following strategies:

  w Affected antennas may be synchronized by a separate user application using a time-multiplexing strategy.

  w Antenna power can be reduced by reconfiguring the RF Transmit Power setting for the Reader.

Note

Performance tests conducted under typical operating conditions at your site are recommended to help you optimize system performance.